UNIVERSIDAD DE LOS ANDES

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

Tesis de Doctorado en Ingeniería

Presentada por:

Juan G. VILLEGAS

VEHICLE ROUTING PROBLEMS WITH TRAILERS

Jurado

Revisores:
Marc SEVAUX — Profesor, Université de Bretagne-Sud
Angel CORBERÁN — Profesor, Universitat de València

Examinadores:
Dominique FEILLET — Profesor, Ecole des Mines de Saint-Etienne
Kenneth SÖRENSEN — Profesor, Université d'Anvers

Asesores:
Christian PRINS — Profesor, Université de Technologie de Troyes
Andrés MEDAGLIA — Profesor Asociado, Universidad de los Andes

Co-asesores:
Caroline PRODHON — Profesor Asistente, Université de Technologie de Troyes
Nubia VELASCO — Profesor Asistente, Universidad de los Andes

Grupos de Investigación: LOSI-Laboratoire d'Optimisation des Systèmes Industriels
COPA-Centro para la Optimización y Probabilidad Aplicada

UNIVERSITÉ DE TECHNOLOGIE DE TROYES

École Doctorale

SCIENCES DES SYSTÈMES TECHNOLOGIQUES ET ORGANISATIONNELS

Année: 2010

Thèse de doctorat de l'Université de Technologie de Troyes

Spécialité : Optimisation et sûreté des systèmes

Présentée et soutenue publiquement par

Juan G. VILLEGAS

VEHICLE ROUTING PROBLEMS WITH TRAILERS

Jury

Rapporteurs:

Marc SEVAUX	Professeur, Université de Bretagne-Sud
Angel CORBERÁN	Professeur, Universitat de València

Examinateurs:

Dominique FEILLET	Professeur, Ecole des Mines de Saint-Etienne
Kenneth SÖRENSEN	Professeur, Université d'Anvers

Directeurs de thèse:

Christian PRINS	Professeur, Université de Technologie de Troyes
Andrés MEDAGLIA	Professeur Associé, Universidad de los Andes

Co-encadrants:

Caroline PRODHON	Maître de conférences, Université de Technologie de Troyes
Nubia VELASCO	Professeur Assistente, Universidad de los Andes

Laboratoires: LOSI-Laboratoire d'Optimisation des Systèmes Industriels
COPA-Centro para la Optimización y Probabilidad Aplicada

Con todo mi amor para Anna y Maria Clara, el principio y fin de todas mis rutas.

Acknowledgements

I first thank my wife, Anna Sofía, and my daughter, María Clara, for their support, sacrifice, company, and love. They were always there, no matter the place, the weather or the language.

I would like to start by acknowledging my advisors, Andrés Medaglia, Christian Prins, Caroline Prodhon, and Nubia Velasco, for their intellectual generosity. I have learned from them many things that go far beyond optimization, vehicle routing, and operations research. This dissertation would not have been possible without the confidence they gave me from the early beginnings. Working with them has been a pleasure.

I would also like to thank Professors José Manuel Belenguer and Enrique Benavent, and Ph.D. student Antonio Martínez, from the Universitat de València, for sharing with me all their knowledge and experience on cutting plane methods. Their hospitality had made our short stay in Valencia unforgettable.

I want to thank Professor Rodrigo Cardozo of Uniandes for the tutorial on data structures and algorithms; it changed the way I think about programming. Life in France was easier thanks to the good French courses I received at the UTT from Cyrille Guérinot and Dominique Masson.

I thank Professors Angel Corberán and Marc Sevaux for taking time out of their busy schedules to read this dissertation and give their feedback; I also want to thank Professors Dominique Feillet and Kenneth Sörensen for accepting the invitation to be part of the jury.

I thank LOSI Professors Farouk Yalaoui and Lionel Amodeo for their kindness every time I wanted to practice my French; I also thank Mohamed Reghioui for sharing with me his experience on some of the metaheuristics I implemented in this thesis. Special thanks to professor Lionel Amodeo for being part of the jury of the proposal. Many thanks to all COPA members, especially to Jose Luis Walteros, I really enjoyed teaching with him the introductory optimization course. I could not forget my good friend and fellow Ph.D. student Jorge Mendoza, I miss the conversations in which we share ideas and discuss plans; I also thank Jose Luis and Jorge for their help with LaTeX and Java. Valentina, Maria Isabel and Juan David deserve special thanks for their friendship and for being there in the hard days.

I thank my parents for being an example to follow, for their unconditional support, and for encouraging me to pursue this degree, even if this separated us for some time. Special thanks to my brother and colleague, Andrés Mauricio, talking with him was always useful when I wanted to clarify my ideas; and living with him the last six months at Bogotá made it easier. Also my greatest gratitude goes to our friends of the GLT (Grupo Latinos en Troyes), Gabriela, Demetrio, Gala, Flavia, Michel, Adrien, Valentin, Sarah, Edwin, Sasha, and Tatiana, for being our family in France.

Pascale, Marie-José, Veronique, Patricia and España deserve a special thank for taking care of many administrative matters over these three years. I would like to thank Ana Maria Velez from Universidad EAFIT (Colombia) for providing me with the space where I prepared the final document during the last months.

I thank the Department of Industrial Engineering and the Ph.D. Program of the Faculty of Engineering at Uniandes, and LOSI at the UTT for the financial support I received to participate in the different conferences. I also want to thank Professor Angel Corberán for the invitation to CORAL 2009, one of the best conferences I have ever been.

Finally, this dissertation would not have been possible without the financial support of the Champagne-Ardenne Regional Council (France), Colciencias - Administrative Department of Science, Technology and Innovation (Colombia), and the Universidad de Antioquia (Colombia), I gratefully acknowledge them for it.

Juan G. Villegas
October, 2010.

Abstract

Vehicle routing problems (VRPs) are a class of combinatorial optimization problems with application in many different domains ranging from the distribution of goods to the delivery of services. In this thesis we have studied VRPs in which the capacity of the vehicles is increased with the use of detachable trailers. This thesis comprises three parts:

The first part is devoted to the single truck and trailer routing problem with satellite depots (STTRPSD), where a single truck with a detachable trailer based at a depot serves the demand of a set of customers accessible only by truck. For this problem we have developed three heuristics, two metaheuristics based on GRASP and evolutionary local search, and an exact branch-and-cut algorithm.

The second part addresses the truck and trailer routing problem (TTRP) that models the multi-vehicle case, where a heterogeneous fixed fleet of trucks and trailers is used to serve the demand of a set of customers, some of them with accessibility restrictions. To solve the TTRP we have developed two methods: (i) a hybrid metaheuristic combining GRASP, variable neighborhood search (VNS) and path relinking; and (ii) a matheuristic that uses the routes of the local optima produced by a GRASP/VNS to solve a set-partitioning formulation of the TTRP with a commercial optimizer.

Finally, the third part presents an object-oriented framework for the rapid prototyping of heuristic methods based on the route-first, cluster-second principle. This framework provides a set of reusable components that can be adapted to tackle different VRP extensions.

Keywords: Operations research, Combinatorial optimization, Heuristics, Transportation, Logistics.

Résumé

Les problèmes de tournées de véhicules forment une classe de problèmes d'optimisation combinatoire avec des applications dans de nombreux domaines comme la distribution de marchandises et l'exécution de services. Cette thèse en trois parties étudie des problèmes de tournées où la capacité des véhicules peut être augmentée par des remorques détachables.

La première partie est consacrée à un problème noté STTRPSD, dans lequel un camion avec une remorque détachable doit visiter à partir d'un dépôt des clients accessibles par le camion sans sa remorque. La remorque doit donc être laissée temporairement sur certains nœuds du réseau. Pour ce problème, nous avons développé trois heuristiques, deux métaheuristiques de type GRASP et recherche locale évolutionnaire, et une méthode exacte de type branchement et coupes.

La deuxième partie traite un problème nommé TTRP, qui étend le STTRPSD à plusieurs véhicules hétérogènes et à des clients accessibles ou non avec les remorques. Pour le résoudre, nous avons conçu deux méthodes : (i) une métaheuristique hybride combinant un GRASP, une recherche à voisinage variable et un *path relinking*; et (ii) une matheuristique qui utilise les optima locaux produits par un GRASP/VNS pour résoudre un problème de partitionnement à l'aide d'un solveur commercial.

Enfin, la troisième partie présente une librairie orientée objet pour le prototypage rapide de méthodes heuristiques basées sur le principe *route-first, cluster-second.* Cette librairie fournit des composants logiciels réutilisables qui peuvent être adaptés pour gérer différentes extensions.

Mots clés: Recherche opérationnelle, Optimisation combinatoire, Heuristique, Transport, Logistique.

Contents

Chapter I

Introduction

Introduction

Transport operations can be found at different stages of supply chains, from the collection of raw materials to the distribution of final products. In European countries, transportation represents 19% to 37% of total logistic cost for several industries [56]. Not surprisingly, operations research has been successfully applied to model and optimize different transportation decisions. Among them, node-routing decisions are faced in different settings such as soft-drink distribution [69,101], sales-force routing [68], milk collection [24], dairy distribution [120], frozen-food distribution [44], meat distribution [15], recycling operations [13,104], school-bus routing [29], postal delivery [121], industrial-waste collection [109], home healthcare [43], ready-mixed concrete delivery [120], maintenance operations in public utilities [81], raw-materials collection [1], final-products delivery [95,125], and after-sales services [20,125], among others. However, they all share the same underlying structure in which a set of mobile servers (usually vehicles) satisfies the demand for a good (or service) of a set of geographically scattered customers. This underlying structure has been captured in the well-known vehicle routing problem (VRP).

Formally, the VRP is defined as an optimization problem in which a set of vehicles of limited capacities based at a depot has to serve the demand of a geographically scattered set of customers. The objective of the VRP is to find a set of routes of minimum total length such that each customer is visited exactly once, all the routes begin and end at the depot and the total demand of the customers visited in each route does not exceed the capacity of the allocated vehicle. Since the seminal paper by Dantzig and Ramser [35], the VRP has attracted the attention of many researchers in the optimization field. For instance, Eksioglu et al. [41] report more than one thousand journal articles addressing VRP-related problems. Moreover, these authors found a significant growth in the number of publications during the last two decades.

As pointed out by Golden et al. [61] the VRP is one of the success stories of operational research. A recent survey of vehicle routing software presents 22 products with customers from different industries [91]. Moreover, a study of different real-world applications estimates that the cost reduction achieved with the use of optimization techniques range from 7% to 37%, depending on the characteristics of the routing problem [116]. Currently, the need to reduce green-house emissions gives new reasons to study the VRP and its extensions [110].

The classical VRP has been extended with side constraints to model several characteristics found in practice. A non-exhaustive list of VRP extensions includes: the distance-constrained VRP in which a maximum limit for the length or travel time of any route exists [73,79]; the VRP with time windows, where each customer must be visited within a prespecified time interval [30]; the VRP with backhauls, where a subset of delivery customers (or *linehaul customers*) must be visited before a subset of collection customers (or *backhaul customers*) [59,108], the VRP with pick-ups and deliveries in which a transportation request has a pick-up and a delivery location [37,89,90]; heterogeneous fleet VRPs that consider a fleet composed of vehicles with different capacities and routing costs [6,98]; the site-dependent VRP in which each customer can be served only by a subset of vehicles from a heterogeneous fleet (*incompatibility constraints*) [26,27]; the open VRP in which the vehicles are not required to return to the depot after completing service [47,77]; the periodic VRP, where customers must be visited several times over a planning horizon that spans a few days [31,49]; the split-delivery VRP, where each customer may be visited by several routes [2,22]; the multi-compartment VRP, where each customer orders several products, the vehicles have several compartments, and each compartment is dedicated to one product [42]; the multi-depot VRP (MDVRP), where vehicles are based at several depots [31,103], the truck and trailer routing problem, where a heterogeneous fleet composed of trucks and trailers has to serve a set of customers with incompatibility constraints [25], among others.

New variants of the VRP also appear when the routing is integrated with other strategic, tactical or operational decisions. For instance, the routing decisions are included when locating depots in the location-routing problem [40,86]; inventory control and routing decisions have to be made simultaneously in the inventory-routing problem [19]; and two- and three-dimensional loading vehicle routing problems [50,52] consider the loading of the vehicles when designing their routes. More recently, the term *rich VRP* has been coined to denote vehicle routing problems that include several real-world features often ignored in academic research [65]. For other extensions of the VRP arising in practical applications the reader is referred to Hasle and Kloster [66].

It is clear that a unique vehicle routing problem does not exist, rather there is a wide family of problems with a common structure. Therefore, good vehicle routing methods are those that produce accurate results in short running times, but also they are easy to code and understand, have few parameters, and easily adjust to a wide variety of side constraints found in practice [32]. Traditionally, the solutions methods for vehicle routing problems have been classified into three groups: (i) exact methods, (ii) heuristics, and (iii) metaheuristics.

Exact methods work over different mathematical formulations of the VRP. For instance, two-index vehicle flow formulations use integer variables to indicate the number of times that a given edge is traversed in the solution. Edges between pairs of customers only take binary values, while edges between the depot and the customers may also take the value of 2 representing back-and-forth trips from the depot to the customers. Since the number of capacity constraints in this formulation is exponential, solution methods based on it usually use branch-and-cut approaches. Moreover, several families of valid inequalities have been developed to strengthen its linear relaxation [84]. Solution methods based on two-index vehicle flow formulations include the earlier work by Laporte et al. [74] and the more recent branch-and-cut algorithm by Lysgaard et al. [80]. Three-index vehicle flow formulations [60] in which the vehicle that traverses the edge is specified have not been as successful as the two-index formulation [72]. Currently, branch-and-cut algorithms based on two-index vehicle flow formulations are the best methods to solve VRPs in which the capacity of the vehicle is large with respect to the demands of the customers. Using this formulation Augerat et al. [4] and Lysgaard et al. [80] solved a VRP with 135 customers, that is the largest non-trivial instance solved by an exact method to date [11]. Likewise, commodity-flow formulations use binary variables to indicate if an edge is used in the solution and continuous variables to represent the load of the vehicle when traversing each edge. Valid inequalities for the two-index formulations are also valid for commodity-flow formulations, and can be used to strengthen its linear relaxation [8].

The VRP can be formulated also as a set-partitioning problem in which the columns correspond to feasible routes [12]. However, using this formulation directly is not possible because of the exponential size of the set of feasible routes. Therefore, exact approaches based on this formulation employ column generation techniques. For instance, Baldacci et al. [7] have presented an exact method based on a set-partitioning formulation with additional cuts capable of solving problems with up to 121 customers. Moreover, the flexibility of their method has been illustrated by Baldacci and Mingozzi [9] and Baldacci et al. [5] with extensions that solve the VRP with heterogeneous fleet, time windows, pick-ups and deliveries, multiple depots, periodicity, and incompatibility constraints. Baldacci et al. [10,11] have reviewed and compared the recent exact methods for the VRP.

Since the VRP is an NP-Hard problem [76], exact methods solve consistently instances with up to one hundred customers [10]. Therefore, heuristics and metaheuristics are used in most practical applications where several hundreds of customers are visited daily [62]. An overview of these approximate methods follows.

Most of the early methods for vehicle routing were simple heuristics intended to find good quality solutions in short times. Laporte and Semet [75] classify them into three groups: (i) constructive methods, (ii) two-phase methods and (iii) improvement heuristics.

Constructive methods merge existing routes using a saving criterion as in the popular

Clarke and Wright heuristic [28], or sequentially add customers to routes using an insertion cost as in the Mole and Jameson heuristic [83].

Two-phase heuristics decompose the VRP in the assignment of customers to routes and the sequencing of customers within routes. Cluster-first, route-second methods first group the customers in routes (clusters) that can be served by a single vehicle and then solve a traveling-salesman problem (TSP) for each route. There are different variants of this approach depending on the method used in the clustering phase. For instance, the sweep heuristic by Gillet and Miller [57] use intuitive geometric procedures to group customers, while the Fisher and Jaikumar heuristic [46] solves a generalized assignment problem in the clustering phase. On the other hand, route-first, cluster-second heuristics [14,99], first construct a TSP tour visiting all the customers and then (optimally) break it into VRP feasible routes using a tour splitting procedure.

Improvement heuristics apply simple local search procedures to explore the neighborhood of a VRP solution. These methods operate over single or multiple routes. In the former case, any TSP improvement heuristic can be used, for example, the classical Or-opt [88], 2-opt [48] and 3-opt procedures [34]. The latter case comprises several edge- and node-exchange procedures; Kindervater and Savelsbergh [70] classified them into node relocation, node exchange and edge crossover. More recently, Funke et al. [51] have reviewed most of the local search operators for vehicle routing problems and have proposed a unified representation that allows the modeling of problems with complex constraints. Local-search heuristics for the VRP evolved in metaheuristics that achieve much better results in somewhat longer, yet reasonable running times.

Vehicle routing problems exhibit an impressive record of successful metaheuristic implementations [54]. We understand by metaheuristic, a high level heuristic procedure designed to guide other methods or processes towards achieving reasonable solutions to difficult mathematical optimization problems. Metaheuristics are particularly concerned with not getting trapped at a local optimum (when multiple local optima exist) and/or judiciously reducing the search space [114]. Metaheuristics include genetic algorithms, simulated annealing, tabu search, variable neighborhood search (VNS), iterated local search (ILS), evolutionary strategies, greedy randomized adaptive search procedures (GRASP), scatter search, ant colony optimization, among others. Currently, hybrid metaheuristics combining components and principles from different metaheuristics provide very effective solution methods for several combinatorial optimization problems [21]. For introductory tutorials of several metaheuristics and other related topics the reader is referred to [53,58].

There is a huge number of metaheuristics for the solution of the classical VRP. Among them, the most successful ones are based on evolutionary strategies [82,97] memetic algorithms [18,85,96], adaptive memory programming [107,119], ant colony optimization [102], tabu search [33,36,117] and adaptive large neighborhood search [94]. Gendreau et al. [54] provide an excellent categorized survey of metaheuristics for the VRP and several of its extensions. More recently, matheuristic approaches combining metaheuristics and exact methods in a cooperative way have emerged as a promising alternative for the solution of different VRP variants, such as the classical VRP [107], the split delivery VRP [3] and the location-routing problem [100], among others. For a recent survey of matheuristics to solve different VRPs the reader is referred to Doerner and Schmid [38].

Nonetheless, many of the most successful vehicle-routing metaheuristics are overfitted to solve efficiently an specific variant, yet they have many components and/or a lot of parameters. This overengineering phenomenon comes at the price of a loss of simplicity and flexibility [72]. Consequently, commercial vehicle routing packages do not include most of the rather standard components of the academic vehicle routing metaheuristics such as memory structures, mutation and crossover operators [115]. Therefore, there is a need for flexible vehicle routing methods capable of solving different VRP variants without many modifications, even if this comes at the price of a reasonable loss in solution quality [72].

For additional reference on the VRP, its modeling, solution methods, extensions, and practical applications the reader is referred to the introductory tutorial by Laporte [71], the historic perspective by Laporte [72] and the books by Toth and Vigo [122] and by Golden et al. [61].

Vehicle routing problems with trailers

In this thesis we studied vehicle routing problems with trailers, that is, a vehicle routing problem in which the capacity of the truck is increased by a trailer. Despite its benefits, trailers cause incompatibility constraints at some customers with limited maneuvering space or accessible through narrow streets. These customers can be served only by truck (after detaching the trailer).

Real-world applications of this type of problems appear in distribution and collection operations in rural areas and crowded cities. For instance, in several European countries milk collection is performed by a small tanker with a removable tank trailer of larger capacity [24,67,123]. Some farms are not reachable by big vehicles, so the tank trailer needs to be detached on main roads before visiting them. Gerdessen [55] reported two applications of vehicle routing problems with trailers in the Netherlands, the first one for the distribution of compound animal feed in rural regions and the second one for the distribution of dairy products. Semet and Taillard [113] described a vehicle routing problem with trailers, time windows, site dependencies and heterogeneous fleet arising in the distribution operations of a chain of grocery stores in Switzerland.

The arc-routing equivalent of vehicle routing problems with trailers arises in the design of park-and-loop routes for postal delivery [78], where the postman drives a vehicle from the postal facility to parking locations, loads his sack, and delivers mails by walking the streets; in this case the postman corresponds to the truck and his vehicle to the trailer. Waste collection in small cities and towns also has a similar structure. For instance, in Due Carrare (Italy) [92] small collection vehicles serve narrow streets and big compactors collect waste on streets without accessibility restrictions. Since the disposal facility is far from the town, small collection vehicles meet big compactors in the middle of the routes to dump their contents avoiding long empty trips.

In spite of its wide applicability, vehicle routing problems with trailers are seldomly studied. For instance, the literature review by Gendreau et al. [54] only reports three papers with metaheuristics addressing this type of problems. Moreover, apart from the Lagrangian relaxation method by Semet [112] and the branch-and-price by Drexl [39], we are not aware of other exact algorithms for vehicle routing problems with trailers. Therefore, in this thesis we studied two vehicle routing problems with trailers, namely, the single truck and trailer routing problem with satellite depots (STTRPSD) and the truck and trailer routing problem (TTRP).

In the STTRPSD, a truck with a detachable trailer based at a main depot has to serve the demand of a set of customers accessible only by truck. Therefore, before serving the customers, it is necessary to detach the trailer in designated parking places (called trailer points or satellite depots) where goods are transferred between the truck and the trailer. A solution of the STTRPSD, is composed of a first-level trip departing from the main depot (performed by the truck with the trailer) visiting a subset of trailer points; and several second-level trips (performed by the truck), rooted at trailer points visited in the first-level trip. The multi-depot VRP can be seen as a special case of the STTRPSD where the distance between any two depots is null. Likewise, the STTRPSD can be seen as a simplified version of the two-echelon capacitated location-routing problem (2E-LRP) [63,87] with only one vehicle in the first echelon and without fixed cost at the depots.

On the other hand, in the TTRP a heterogeneous fixed fleet of trucks and trailers serve the demand of a set customers. The customers are partitioned into *truck customers* and *vehicle customers*. Truck customers have incompatibility constraints, being accessible only by truck. In contrast, vehicle customers do not have accessibility restrictions, and their locations can be used to park the trailer before serving truck customers. In the TTRP there are three types of routes: *pure truck routes* performed by a truck; *pure vehicle routes* performed by a truck with a trailer serving only vehicle customers; and *vehicle routes with subtours* performed by a truck with a trailer and serving both vehicle customers and truck customers. Vehicle routes with subtours have a STTRPSD-like structure, then the TTRP generalizes in some sense the STTRPSD. The TTRP is also related with the heterogeneous fixed fleet VRP [6,118] because there are two types of vehicles with different capacities (the truck and the truck-with-the-trailer). In the same way, the TTRP is related with the site-dependent VRP because of the incompatibility constraints of truck customers [26].

In this thesis, three complex variants of the VRP (STTRPSD, MDVRP and TTRP) were tackled successfully using route-first, cluster-second procedures. Thus, the last part is devoted to the development of a simple and flexible framework for the solution of vehicle routing problems with route-first, cluster-second heuristics and metaheuristics. Extracting the common elements of the tour splitting procedures it was possible to develop an object-oriented framework, where most of the logic of the route-first, cluster-second procedure is already included. Using this software framework users tackling real-world VRPs can reduce the coding effort to produce prototypical methods to solve their problems.

This document comprises three parts: the first part is devoted to metaheuristics and exact methods for the STTRPSD; the second part presents hybrid metaheuristics and matheuristics to solve the TTRP; and the third part presents the object-oriented framework for the development of route-first cluster-second heuristics, including one for the TTRP. The document is composed of five self-contained research papers written in collaboration with coauthors who are mentioned at the beginning of each paper. Being self-contained, each paper presents its own introduction, notation, conclusions, and references. In addition, the last chapter presents global conclusions and future research perspectives. Appendix A summarizes the publications developed during the preparation of the thesis and Appendix B presents a summary of the thesis in French. A brief description of the contributions of each chapter follows.

Chapter II: GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots

This chapter introduces the single truck and trailer routing problem. Given that the STTRPSD has never been tackled before, this chapter presents an integer programming formulation to define it. Since the STTRPSD generalizes the MDVRP and the VRP, this chapter describes several heuristics and metaheuristics to solve it. The first heuristic is an intuitive cluster-first route-second approach in which customers are assigned to the closest satellite depot and a best insertion heuristic is used to construct first- and second-level trips. The next heuristic is a route-first cluster-second method. It obtains STTRPSD solutions by optimally splitting TSP tours that only visit customers. A dynamic programming procedure developed for the optimal splitting of giant tours simultaneously builds second-level trips, selects satellite depots and produces the first-level trip. The third heuristic is a variable neighborhood descent (VND) procedure [64] with five neighborhoods, the first three are intended to improve the routing within the trips while the two other improve the first-level trip by adding or dropping satellite depots.

The route-first, cluster-second procedure and the VND are the building blocks of two metaheuristics. The first one is a hybrid GRASP/VND that uses VND as an improvement procedure [105]; while the second one is a multi-start evolutionary local search, in which an evolutionary local search [126] is restarted from different initial solutions obtained by strongly perturbing the giant tour of the current best solution. This method alternates between giant tours and solutions: while the mutation and perturbation are performed over giant tours, the VND operates over STTRPSD solutions.

Being a new problem, there are no publicly available test instances of the STTRPSD. Indeed, we generated a test bed of 32 problems with different characteristics publicly available at `http://hdl.handle.net/1992/1125`. The computational experiments performed over this test bed showed that the multi-start evolutionary local search outperforms GRASP/VND both in terms of solution quality and speed. Moreover, we also tested the metaheuristics developed for the STTRPSD on the MDVRP. Among them, the multi-start evolutionary local search achieves competitive results.

Preliminary versions of the methods described in this chapter were presented in two international conferences:

– J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia and N. Velasco. Metaheuristics for a truck and trailer routing problem. In *CORAL 2009: III Combinatorial Optimization, Routing and Location Meeting*, Elche (Spain), June 10-12, 2009.

– J.G. Villegas, A.L. Medaglia, C. Prins, C. Prodhon, and N. Velasco. GRASP/evolutionary local search hybrids for a truck and trailer routing problem. In *MIC 2009: The VIII Metaheuristics International Conference*, Hamburg (Germany), July 13-16, 2009.

This chapter has been published in Engineering Applications of Artificial Intelligence.

The complete reference follows:

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794, 2010.

Chapter III: A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots

After having presented effective metaheuristics for the STTRPSD in Chapter II, this chapter describes an exact method to solve it. First, we presented a new two-index vehicle flow formulation suitable for the solution with a cutting plane approach. Then, to strengthen its linear relaxation, we introduced several families of valid inequalities, some of them were adapted to the STTRPSD from the capacitated vehicle routing problem [80], the location-routing problem [16], and the multi-depot multiple TSP [17]; and some others were derived exclusively to the STTRPSD. This chapter presents several exact and heuristic procedures for the separation of violated inequalities. Using this formulation and separation procedures we developed a branch-and-cut algorithm for the STTRPSD.

Computational experiments, on the test bed introduced in Chapter II, showed that a partial branch-and-cut, with only the first-level trip variables defined as integer, provides good lower bounds. Moreover, the full branch-and-cut algorithm is capable of solving (in less than 15 minutes) instances with up to 50 customers and 10 satellite depots. The branch-and-cut procedure also succeeded in the solution of clustered problems with up to 100 customers within a time limit of 3 hours. Complementing the good results of Chapter II, for all the problems solved via branch-and-cut the optimality of the best-known solution found with the multi-start evolutionary local search was proved.

Currently, we are improving the general structure of the method, and looking for new valid inequalities in order to solve larger instances. Preliminary results of the branch-and-cut algorithm have been presented in the following conferences:

– J.G. Villegas, J. M. Belenguer, E. Benavent, A. Martínez, C. Prins, and C. Prodhon. A cutting plane approach for the single truck and trailer routing problem with satellite depots. In *EURO 2010: XXIV European Conference on Operational Research*, Lisbon (Portugal), July 11-14, 2010.

– J. M. Belenguer, E. Benavent, A. Martínez, C. Prins, C. Prodhon, and J.G. Villegas. A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots. In *SEIO 2010: XXXII Congreso Nacional de Estadística e Investigación Operativa*, A Coruña (Spain), September 14-17, 2010.

Chapter IV: A GRASP with evolutionary path relinking for the truck and trailer routing problem

This chapter presents a hybrid metaheuristic for the TTRP that effectively combines elements of GRASP [45], VNS [64] and path relinking (PR) [106]. In contrast to most of the previous methods that solve the TTRP based on cluster-first, route-second procedures [23,25,111], this chapter introduces a new route-first, cluster-second procedure.

The randomized construction phase of a hybrid GRASP/VNS is performed with a route-first, cluster-second heuristic, whereas a VNS procedure is used for the improvement phase. The VNS explores four neighborhoods intended to improve all the routes and sub-tours, and a specialized TTRP neighborhood to improve the structure of STTRPSD-like routes. Since our GRASP/VNS algorithm is allowed to explore feasible and infeasible solutions, the VNS procedure serves both as an improvement procedure and as a reparation operator. Additionally, a path relinking procedure has been included to improve the results of the hybrid GRASP/VNS. Three different alternatives were tested for PR: as post-optimization procedure, as intensification procedure, and in an evolutionary path relinking procedure (EvPR).

Computational experiments on the test bed introduced by Chao [25] unveiled the contribution of PR to the quality of solutions. All the GRASP/VNS with PR variants outperform the previous methods from the literature as well as a simple GRASP/VNS without PR. The use of path relinking as post-optimization procedure offers a good tradeoff between solution quality and running time; whereas, the best results were found with EvPR, yet in longer running times. Moreover, GRASP/VNS with PR was able to improve 4 out of 21 best-known solutions and is currently the best published method for the TTRP.

A preliminary version of this chapter was presented at TRISTAN VII:

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND with path relinking for the truck and trailer routing problem. In *TRISTAN VII: Seventh Triennial Symposium on Transportation Analysis*, Tromsø (Norway), June 20-25, 2010.

This chapter has been accepted for publication in Computers & Operations Research, and is available at:

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP with evolutionary path relinking for the truck and trailer routing problem. Doi: 10.1016/j.cor.2010.11.011, 2010.

Chapter V: A matheuristic for the truck and trailer routing problem

Motivated by the results of the previous chapter, where GRASP/VNS arises as a good alternative to generate diverse high-quality solutions for post-optimization procedures, this chapter presents a matheuristic combining GRASP/VNS and integer programming.

This chapter introduces a set-partitioning formulation of the TTRP that is used in a two stage matheuristic. In a first stage, a pool of columns (routes) is built by extracting the routes of the local optima found by GRASP/VNS; then, a second stage tries to derive a better solution by solving the set-partitioning problem over the routes in the pool.

This matheuristic outperforms the GRASP/VNS with path relinking of the previous chapter with comparable running times. Moreover, seven new best-known solutions were found by the proposed matheuristic. Nonetheless, there is still room for improvement. We are currently exploring specialized methods to solve the set-partitioning problem and pool-management strategies to reduce the size of the pool.

This chapter has been submitted for presentation at IESM 2011 (International Conference on Industrial Engineering and Systems Management) that will take place in Metz (France) in May 2011.

Chapter VI: A route-first cluster-second computational framework for vehicle routing heuristics

During the development of the heuristic and metaheuristic methods of the previous chapters, we tackled different VRP variants using route-first, cluster-second procedures with competitive results. Moreover, in the VRP community there is a need for simple and flexible methods to solve VRPs with different side constraints. Thus, this chapter presents an extensible computational object-oriented framework for rapid prototyping of heuristic methods based on the route-first cluster-second principle. More importantly, it provides the user wide a set of reusable components that can be adapted to tackle his/her specific VRP variant with a reasonable coding effort.

The flexibility of the framework is illustrated through a simple evolutionary strategy to solve the capacitated VRP and the TTRP introduced in Chapter IV. Although, this evolutionary strategy is not intended to be the best method for the TTRP or the VRP (yet a simple one), in both problems it achieves results as good as those of specialized constructive and local search procedures. The framework is publicly available at `http://copa.uniandes.edu.co/?p=181` and has been tested on two real-world applications with good results [93,124].

The architecture of the framework and the examples have been presented in the following international conferences:

– J. G. Villegas, N. Velasco, C. Prins, J. E. Mendoza, and A. L. Medaglia. A split-based evolutionary framework for vehicle routing. In *IERC 2008: Industrial Engineering Research Conference*, Vancouver (Canada), May 17-21, 2008.

– J. G. Villegas, A. L. Medaglia, J. E. Mendoza, C. Prins, C. Prodhon, and N. Velasco. A split-based framework for the vehicle routing problem. In *CLAIO 2008: XIV Congreso Latino Ibero Americano de Investigación de Operaciones*, Cartagena (Colombia), September 9-12, 2008.

– J. G. Villegas, A. L. Medaglia, C. Prins, C. Prodhon, and N. Velasco. Solving the truck and trailer routing problem with a route-first, cluster-second framework. In *ALIO/INFORMS Joint International Meeting*, Buenos Aires (Argentina), June 6-9, 2010.

References

[1] J. Alegre, M. Laguna, and J. Pacheco. Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *European Journal of Operational Research*, 179(3):736–746, 2007.

[2] C. Archetti and M. G. Speranza. The split delivery vehicle routing problem: a survey. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 103–122. Springer, 2008.

[3] C. Archetti, M. G. Speranza, and M. W. P. Savelsbergh. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, 42(1):22–31, 2008.

[4] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report RR- 949-M, ARTEMIS-IMAG, Université. Joseph Fourier, Grenoble, 1995.

[5] R. Baldacci, E. Bartolini, A. Mingozzi, and R. Roberti. An exact solution framework for a broad class of vehicle routing problems. *Computational Management Science*, 7(3):229–268, 2010.

[6] R. Baldacci, M. Battarra, and D. Vigo. Routing a heterogeneous fleet of vehicles. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 3–27. Springer, 2008.

[7] R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, 2007.

[8] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, 52(5):723–738, 2004.

[9] R. Baldacci and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380, 2008.

[10] R. Baldacci, P. Toth, and D. Vigo. Recent advances in vehicle routing exact algorithms. *4OR-A Quarterly Journal of Operations Research*, 5:269–298, 2007.

[11] R. Baldacci, P. Toth, and D. Vigo. Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, 175(1):213–245, 2009.

[12] M. L. Balinski and R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2):300 – 304, 1964.

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

[14] J. E. Beasley. Route-first cluster-second methods for vehicle routing. *Omega*, 11:403–408, 1983.

[15] J. M. Belenguer, E. Benavent, and M. Martinez. Rutarep: a computer package to design dispatching routes in the meat industry. *Journal of Food Engineering*, 70:435–445, 2005.

[16] J. M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Wolfler-Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 2010. doi: 10.1016/j.cor.2010.09.019.

[17] E. Benavent and A. Martinez. A branch-and-cut algorithm for the multi depot multiple TSP. In *TRISTAN VII: Seventh Triennial Symposium on Transportation Analysis*, pages 70–73, Tromsø, Norway, June 20-25, 2010.

[18] J. Berger and M. Barkaoui. A new hybrid genetic algorithm for the capacitated vehicle routing problem. *Journal of the Operational Research Society*, 54(12):1254–1262, 2003.

[19] L. Bertazzi, M. Savelsbergh, and M. G. Speranza. Inventory routing. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 49–72. Springer, 2008.

[20] F. Blakeley, B. Bozkaya, B. Cao, W. Hall, and J. Knolmajer. Optimizing periodic maintenance operations for Schindler Elevator Corporation. *Interfaces*, 33:67–79, 2003.

[21] C. Blum and A. Roli. Hybrid metaheuristics: an introduction. In C. Blum, M. J. B. Aguilera, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 114 of *Studies in Computational Intelligence*, pages 1–30. 2008.

[22] M. Boudia, C. Prins, and M. Reghioui. An effective memetic algorithm with population management for the split delivery vehicle routing problem. In T. Bartz-Beielstein, M. J. B. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 16–30, 2007.

[23] M. Caramia and F. Guerriero. A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society*, 61:1168–1180, 2010.

[24] M. Caramia and F. Guerriero. A milk collection problem with incompatibility constraints. *Interfaces*, 40(2):130–143, 2010.

[25] I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29:33–51, 2002.

[26] I.-M. Chao, B. Golden, and E. Wasil. A computational study of a new heuristic for the site-dependent vehicle routing problem. *INFOR*, 37:319–336, 1999.

[27] I.-M. Chao and T.-S. Liou. A new tabu search heuristic for the site-dependent vehicle routing problem. In B. L. Golden, S. Raghavan, and E. A. Wasil, editors, *The Next Wave in Computing, Optimization, and Decision Technologies*, pages 107– 119. Springer, 2005.

[28] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivering points. *Operations Research*, 12:568–581, 1964.

[29] A. Corberán, E. Fernández, M. Laguna, and R. Martí. Heuristic solutions to the problem of routing school buses with multiple objectives. *Journal of the Operational Research Society*, 53:427–435, 2002.

[30] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis. VRP with time windows. In P. Toth and D. Vigo, editors, *The vehicle routing problem*, pages 157 – 193. SIAM, 2002.

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

[32] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53:512–522, 2002.

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

[34] G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, 1958.

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

[36] U. Derigs and R. Kaiser. Applying the attribute based hill climber heuristic to the vehicle routing problem. *European Journal of Operational Research*, 177(2):719–732, 2007.

[37] G. Desaulniers, J. Desrosiers, A. Erdmann, M. M. Solomon, and F. Soumis. VRP with pickup and delivery. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 225–242. SIAM, 2002.

[38] K. F. Doerner and V. Schmid. Survey: Matheuristics for rich vehicle routing problems. In M. J. Blesa, C. Blum, G. R. Raidl, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 6373 of *Lecture Notes in Computer Science*, pages 206–221, 2010.

[39] M. Drexl. A branch-and-price algorithm for the truck-and-trailer routing problem. Technical report, Deutsche Post Endowed Chair of Optimization of Distribution Networks - RWTH Aachen University, 2006.

[40] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon. A GRASP X ELS approach for the capacitated location-routing problem. *Computers & Operations Research*, 37(11):1912–1923, 2010.

[41] B. Eksioglu, A. V. Vural, and A. Reisman. The vehicle routing problem: a taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.

[42] A. El-Fallahi, C. Prins, and R. Wolfler-Calvo. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, 35:1725–1741, 2008.

[43] P. Eveborn, M. Ronnqvist, H. Einarsdottir, M. Eklund, K. Liden, and M. Almroth. Operations research improves quality and efficiency in home care. *Interfaces*, 39(1):18–34, 2009.

[44] J. Faulin, P. Sarobe, and J. Simal. The DSS LOGDIS optimizes delivery routes for FRILAC's frozen products. *Interfaces*, 35:202–214, 2005.

[45] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[46] M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11:109–124, 1981.

[47] K. Fleszar, I. H. Osman, and K. S. Hindi. A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 195(3):803 – 809, 2009.

[48] M. M. Flood. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

[49] P. M. Francis, K. R. Smilowitz, and M. Tzur. The period vehicle routing problem and its extensions. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 73–102. Springer, 2008.

[50] G. Fuellerer, K. F. Doerner, R. F. Hartl, and M. Iori. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, 201(3):751 – 759, 2010.

[51] B. Funke, T. Grünert, and S. Irnich. Local search for vehicle routing and scheduling problems: review and conceptual integration. *Journal of Heuristics*, 11(4):267–306, 2005.

[52] M. Gendreau, M. Iori, G. Laporte, and S. Martello. A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1):4–18, 2008.

[53] M. Gendreau and J.-Y. Potvin, editors. *Handbook of Metaheuristics*. Springer, 2 edition, 2010. In press.

[54] M. Gendreau, J.-Y. Potvin, O. Bräysy, G. Hasle, and A. Løkketangen. Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 143–170. Springer, 2008.

[55] J. C. Gerdessen. Vehicle routing problem with trailers. *European Journal of Operational Research*, 93:135–147, 1996.

[56] G. Ghiani, G. Laporte, and R. Musmanno. *Introduction to Logistics Systems Planning and Control*. Wiley, 2004.

[57] B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

[58] F. Glover and G. A. Kochenberger, editors. *Handbook of Metaheuristics*. Kluwer, 1 edition, 2002.

[59] M. Goetschalckx and C. Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42:39–51, 1989.

[60] B. L. Golden, T. L. Magnanti, and H. Q. Nguyen. Implementing vehicle routing algorithms. *Networks*, 7(2):113–148, 1977.

[61] B. L. Golden, S. Raghavan, and E. Wasil. *The Vehicle routing problem: latest advances and new Challenges*. Springer, 2008.

[62] B. L. Golden, E. Wasil, J. P. Kelly, and I.-M. Chao. The impact of metaheuristics on solving the vehicle routing problem. In T. Crainic and G. Laporte, editors, *Fleet management and logistics*, pages 33–56. Springer, 1998.

[63] J. Gonzalez-Feliu. The N-echelon location routing problem: concepts and methods for tactical and operational planning. Technical Report halshs-00422492, Laboratoire d ‘Economie des Transports, Institut des Sciences de l‘Homme, 2009.

[64] P. Hansen and N. Mladenovic. Variable neighbourhood search: principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

[65] R. Hartl, G. Hasle, and G. Janssens. Special issue on rich vehicle routing problems. *Central European Journal of Operations Research*, 14:103–104, 2006.

[66] G. Hasle and O. Kloster. Industrial vehicle routing. In G. Hasle, K.-A. Lie, and E. Quak, editors, *Geometric Modelling, Numerical Simulation, and Optimization*, pages 397–435. Springer, 2007.

[67] A. Hoff and A. Løkketangen. A tabu search approach for milk collection in western Norway using trucks and trailers. In *TRISTAN VI: The Sixth Triennial Symposium on Transportation Analysis*, Phuket, Tailand, June 10-15, 2007.

[68] W. Jang, H. H. Lim, T. J. Crowe, G. Raskin, and T. E. Perkins. The Missouri lottery optimizes its scheduling and routing to improve efficiency and balance. *Interfaces*, 36(4):302–313, 2006.

[69] G. Kant, M. Jacks, and C. Aantjes. Coca-Cola enterprises optimizes vehicle routes for efficient product delivery. *Interfaces*, 38(1):40–50, 2008.

[70] G. A. P. Kindervater and M. W. P. Savelsbergh. Vehicle routing: handling edge exchanges. In E. H. L. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 337–360. Wiley, 1997.

[71] G. Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics*, 54:811 – 819, 2007.

[72] G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

[73] G. Laporte, M. Desrochers, and Y. Nobert. Two exact algorithms for the distance-constrained vehicle routing problem. *Networks*, 14(1):161–172, 1984.

[74] G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050–1073, 1985.

[75] G. Laporte and F. Semet. Classical heuristics for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 109–128. SIAM, 2002.

[76] J. K. Lenstra and A. H. G. Rinnooy-Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–228, 1981.

[77] A. N. Letchford, J. Lysgaard, and R. W. Eglese. A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, 58(12):1642–1651, 2006.

[78] L. Levy and L. Bodin. Scheduling of local delivery carriers for the United States Postal Service. In M. Dror, editor, *Arc Routing: Theory, Solutions and Applications*, pages 419–442. Kluwer, 2000.

[79] C.-L. Li, D. Simchi-Levi, and M. Desrochers. On the distance constrained vehicle routing problem. *Operations Research*, 40(4):790–799, 1992.

[80] J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, 2004.

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

[82] D. Mester and O. Bräysy. Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & Operations Research*, 34(10):2964 – 2975, 2007.

[83] R. H. Mole and S. R. Jameson. A sequential route-building algorithm employing a generalised savings criterion. *Operational Research Quarterly*, 27(2):503–511, 1976.

[84] D. Naddef and G. Rinaldi. Branch-and-cut algorithms for the capacitated VRP. In P. Toth and D. Vigo, editors, *The vehicle routing problem*, pages 53 – 84. SIAM, 2002.

[85] Y. Nagata and O. Bräysy. Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks*, 54(4):205–215, 2009.

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

[87] V.-P. Nguyen, C. Prins, and C. Prodhon. GRASP with learning process for a two-echelon location routing problem. In *8th International Conference of Modeling and Simulation - MOSIM'10*, Hammamet - Tunisia, May 10-12, 2010.

[88] I. Or. *Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking.* PhD thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL., 1976.

[89] S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems: Part I: transportation between customers and depot. *Journal für Betriebswirtschaft*, 58(1):21–51, 2008.

[90] S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems: Part II: transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58(2):81–117, 2008.

[91] J. Partyka and R. Hall. Vehicle routing software survey. OR/MS Today, February 2010. `http://www.lionhrtpub.com/orms/surveys/Vehicle_Routing/vrss.html`.

[92] A. Pia and C. Filippi. A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, 13(2):125–141, 2006.

[93] C. Pineda. Optimización al problema de entrega de repuestos a la red de concesionarios Sofasa - Bogotá. Master's thesis, Departamento de Ingeniería Industrial Universidad de los Andes, 2009. (In Spanish).

[94] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research.*, 34:2403–2435, 2007.

[95] C. Prins. Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case. *Journal of Mathematical Modelling and Algorithms*, 1:135–150, 2002.

[96] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31:1985–2002, 2004.

[97] C. Prins. A GRASP X Evolutionary Local Search hybrid for the vehicle routing problem. In F. B. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, pages 35–53. Springer, 2009.

[98] C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22(6):916–928, 2009.

[99] C. Prins, N. Labadi, and M. Reghioui. Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research*, 47(2):507–535, 2009.

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

[101] J. Privé, J. Renaud, F. F. Boctor, and G. Laporte. Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society*, 57:1045–1052, 2006.

[102] M. Reimann, K. Doerner, and R. F. Hartl. D-Ants: Savings Based Ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, 31(4):563–591, 2004.

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

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

[105] M. G. C. Resende. Metaheuristic hybridization with greedy randomized adaptive search procedures. In Z.-L. Chen and S. Raghavan, editors, *TutORials in Operations Research*, pages 295–319. INFORMS, 2008.

[106] M. G. C. Resende, C. C. Ribeiro, F. Glover, and R. Martí. Scatter search and path-relinking: fundamentals, advances, and applications. In M. Gendreau and J. Y. Potvin, editors, *Handbook of Metaheuristics*, pages 87–107. Springer, 2010.

[107] Y. Rochat and E. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.

[108] S. Ropke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–775, 2006.

[109] S. Sahoo, S. Kim, B.-I. Kim, B. Kraas, and A. Popov Jr. Routing optimization for waste management. *Interfaces*, 35(1):24–36, 2005.

[110] A. Sbihi and R. W. Eglese. Combinatorial optimization and green logistics. *4OR-A Quarterly Journal of Operations Research*, 5(2):99–116, 2007.

[111] S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33:894–909, 2006.

[112] F. Semet. A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research*, 61:45–65, 1995.

[113] F. Semet and E. Taillard. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research*, 41:469–488, 1993.

[114] E. A. Silver. An overview of heuristic solution methods. *Journal of the Operational Research Society*, 55(9):936–956, 2004.

[115] K. Sörensen, M. Sevaux, and P. Schittekat. Multiple neighbourhood search in commercial VRP packages: evolving towards self-adaptive methods. In C. Cotta, M. Sevaux, and K. Sörensen, editors, *Adaptive and Multilevel Metaheuristics*, volume 136 of *Studies in Computational Intelligence*, pages 239–253. Springer, 2008.

[116] C. Sutcliffe and J. Board. The ex-ante benefits of solving vehicle-routeing problems. *Journal of the Operational Research Society*, 42(2):135–143, 1991.

[117] E. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8):661–673, 1993.

[118] E. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO*, 33(1):1–14, 1999.

[119] C. D. Tarantilis. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research*, 32(9):2309–2327, 2005.

[120] C. D. Tarantilis and C. T. Kiranoudis. A flexible adaptive memory-based algorithm for real-life transportation operations: two case studies from dairy and construction sector. *European Journal of Operational Research*, 179:806–822, 2007.

[121] C. D. Tarantilis, C. T. Kiranoudis, and N. C. Markatos. Use of the BATA algorithm and MIS to solve the mail carrier problem. *Applied Mathematical Modelling*, 26:481–500, 2002.

[122] P. Toth and D. Vigo. *The Vehicle Routing Problem.* SIAM, Philadelphia, 2002.

[123] R. Vahrenkamp. Transportation logistic in rural setting. Technical Report 5/1989, Department of Economics, University of Kassel, 1989.

[124] M. A. Vargas, J. L. Walteros, and A. L. Medaglia. Hybrid approach for airline crew pairing optimization for small carriers. In *IERC 2009: Industrial Engineering Research Conference*, Miami (US), May 30 - June 3, 2009.

[125] D. Weigel and B. Cao. Applying GIS and OR techniques to solve Sears technician-dispatching and home delivery problems. *Interfaces*, 29(1):112–130, 1999.

[126] S. Wolf and P. Merz. Evolutionary local search for the super-peer selection problem and the p -hub median problem. In T. Bartz-Beielstein, M. J. B. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 1–15, 2007.

PART I: Single truck and trailer routing problem with satellite depots

Chapter II

GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots

GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots

Juan G. Villegas[a,b,c] Christian Prins[a] Caroline Prodhon[a] Andrés L. Medaglia[b] Nubia Velasco[b]

[a]*Laboratoire d'Optimisation des Systèmes Industriels (LOSI), Institut Charles Delaunay, Université de Technologie de Troyes. BP 2060, 10010 Troyes Cedex, France*

[b]*Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes. A.A. 4976, Bogotá D.C., Colombia*

[c]*Departamento de Ingeniería Industrial, Universidad de Antioquia. A.A. 1226, Medellin, Colombia*

Abstract

In the single truck and trailer routing problem with satellite depots (STTRPSD) a vehicle composed of a truck with a detachable trailer serves the demand of a set of customers reachable only by the truck without the trailer. This accessibility constraint implies the selection of locations to park the trailer before performing the trips to the customers. We propose two metaheuristics based on greedy randomized adaptive search procedures (GRASP), variable neighborhood descent (VND) and evolutionary local search (ELS) to solve this problem. To evaluate these metaheuristics we test them on a set of 32 randomly generated problems. The computational experiment shows that a multi-start evolutionary local search outperforms a GRASP/VND. Moreover, it obtains competitive results when applied to the multi-depot vehicle routing problem (MDVRP), that can be seen as a special case of the STTRPSD.

Key words: Truck and trailer routing problem (TTRP), multi-depot vehicle routing problem (MDVRP), greedy randomized adaptive search procedures (GRASP), evolutionary local search (ELS), variable neighborhood descent (VND), iterated local search (ILS)

Email addresses: jg.villegas64@uniandes.edu.co, juan_guillermo.villegas@utt.fr, jvillega@udea.edu.co (Juan G. Villegas), christian.prins@utt.fr (Christian Prins), caroline.prodhon@utt.fr (Caroline Prodhon), amedagli@uniandes.edu.co (Andrés L. Medaglia), nvelasco@uniandes.edu.co (Nubia Velasco).

1. Introduction

This chapter presents the single truck and trailer routing problem with satellite depots (STTRPSD), a generalization of the vehicle routing problem (VRP). The VRP is a well known combinatorial optimization problem that aims to find a set of routes of minimum total length to serve the demand of a set of customers using a homogeneous fleet of capacitated vehicles based at a main depot [24].

In the STTRPSD a single vehicle (a truck with a detachable trailer) based at a main depot serves the demand of a set of customers, reachable only by the truck without the trailer. Therefore, there is a set of parking locations (called trailer points or satellite depots) where it is possible to detach the trailer and to transfer products between the truck and the trailer. In feasible solutions of the STTRPSD, each client is assigned to one trailer point. Consequently, trailer points with assigned customers are said to be open. The first-level trip departing from the main depot is performed by the truck with the trailer and visits the subset of open trailer points. Each customer must be served by exactly one second-level trip (performed by the truck alone), starting and ending at the allocated trailer point. Thus, the total load in a second-level trip should not exceed the truck capacity. The goal of the STTRPSD is to minimize the total length of the trips.

The STTRPSD is NP-hard since it includes the VRP (one satellite depot) and the Multi-Depot VRP (null cost between any two depots) as particular cases. Since exact approaches solve consistently VRPs up to only 100 customers [3], the preferred solutions methods are mainly heuristics [25] or metaheuristics [17]. VRP variants appear by incorporating new constraints (e.g., time windows [10]) or by integrating the design of the routes with higher level decisions (e.g. location-routing problems [29]). The books by Toth and Vigo [38] and Golden et al. [19] provide a wide overview of the VRP, its extensions, solution methods, and practical applications.

The STTRPSD is a relevant problem that appears for instance in milk collection, where customers (farms) are often served by a single tanker with a removable tank trailer. Trailer points are in general parking locations on main roads, while farms are located on narrow roads inaccessible with the trailer. In reality, there can be several vehicles but usually farms are clustered based on their geographical location [8], each cluster being assigned to one vehicle. Figure 1 depicts a feasible solution of the STTRPSD composed of: (i) a first-level trip departing from the main depot (performed by the truck with the trailer)

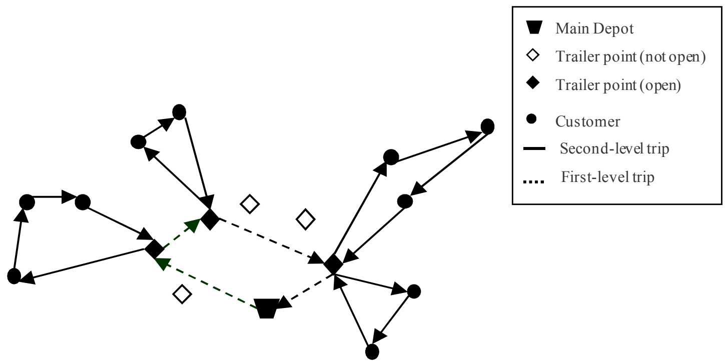


Fig. 1. Feasible solution for the STTRPSD.

visiting a subset of trailer points; and (ii) several second-level trips performed by the truck, starting and ending at the trailer points visited in the first-level trip and visiting subsets of customer whose total demand does not exceed the capacity of the truck.

Another suitable application for the arc routing counterpart of the STTRPSD appears in the design of park-and-loop routes for postal delivery [26], where the postman drives a vehicle from the postal facility to a parking location, loads his sack, and delivers mail by walking the streets forming a loop. Then, he returns to the vehicle, loads his sack again, and delivers by foot in a second loop. Once he has served all the walking loops nearby, he drives the vehicle to the next parking location. At the end of the day the postman returns to the postal facility.

The STTRPSD is also related to the capacitated arc routing problem with intermediate facilities (CARPIF) [18]. The CARPIF is a variant of the capacitated arc routing problem (CARP) with a single vehicle but multiple trips, in which the vehicle is unloaded at intermediate facilities between two consecutive trips. Applications of the CARPIF appear in waste collection where intermediate facilities are dump sites or incinerators.

It is possible to see the STTRPSD as a variant of the truck and trailer routing problem (TTRP), another extension of the VRP in which a heterogeneous fleet composed of m trucks and b trailers ($b < m$) is used to satisfy the demand of a set of customers partitioned in *vehicle* and *truck customers*, where the latter are only accessible by truck. The TTRP has been tackled using tabu search [7,37], simulated annealing [27] and a mathematical-programming based heuristic [6]. The STTRPSD differs from the TTRP in the use of a single vehicle and the definition of trailer points independent from customer locations. Also, when the STTRPSD is extended to include several vehicles and capacitated satellite depots it transforms into the seldomly studied two-echelon capacitated vehicle routing problem (2E-CVRP) introduced by Gonzalez-Feliu et al. [20].

The STTRPSD reduces to the multi-depot VRP (MDVRP) when all the distances between satellite depots are null. In the MDVRP, the fleet of vehicles is based at several depots, and the customers must be visited by exactly one route starting and ending at one of the depots. Recently, Baldacci and Mingozzi [2] proposed a unified exact method capable of solving different variants of the VRP, including the MDVRP. Tabu search [11] and adaptive large scale neighborhood search (ALNS) [30] provide the best results among the metaheuristics developed for the MDVRP. Crevier et al. [13] extended the MDVRP allowing routes that may depart and end at different depots, giving rise to the multi-depot vehicle routing problem with inter-depot routes (MDVRPI).

Hoff and Løkketangen [22] described a practical application of vehicle routing for milk collection in Norway in which the routes have the structure of a STTRPSD. For their particular problem they use tabu search to design the milk collection routes of a set of farms to supply various dairy plants. However, to the best of our knowledge, the STTRPSD has not been formally described in the literature, and no solution algorithm has been designed for it.

The remainder of this chapter is organized as follows. Section 2 presents an integer programming formulation of the STTRPSD. Section 3 describes two metaheuristics based on greedy randomized adaptive search procedures and evolutionary local search. Section 4 presents a computational evaluation of the proposed methods. Finally, we conclude and outline future work in Section 5. Appendix A summarizes the notation used through the chapter.

2. Integer programming formulation

The STTRPSD is defined on a graph $G = (V, A)$, where $V = \{0\} \cup V_D \cup V_C$ is the node set with the main depot at node 0, $V_D = \{1, 2, \ldots, p\}$ is the set of trailer points (satellite depots), and $V_C = \{p+1, p+2, \ldots, p+n\}$ is the set of customers with known demands q_i ($i \in V_C$). A is the arc set, with costs c_{ij} ($i, j \in V$, $i \neq j$) satisfying the triangle inequality. The parameters Q_V and Q_T are the capacities of the truck and the trailer, respectively. To ensure feasibility with one vehicle, the sum of demands does not exceed $Q_V + Q_T$.

To formulate the STTRPSD we define the following subsets of V: (i) $V_1 = V_D \cup \{0\}$, set of nodes that can be visited in the first-level trip; (ii) $V_2 = V_C \cup V_D$, set of nodes that can be visited in second-level trips; and (iii) $V^j = V_C \cup \{j\}$ ($\forall j \in V_D$), set of nodes that can be visited in a second-level trip rooted at satellite depot j. The integer programming formulation of the STTRPSD uses binary variables y_{ij} equal to 1 if and only if the truck with the trailer traverses the arc (i, j) $(i, j \in V_1)$; and binary variables x^j_{lm} equal to 1 if and only if arc (l, m) $(l, m \in V^j)$ is traversed by the truck in a second-level trip departing from trailer point $j \in V_D$. The integer programming formulation of the STTRPSD follows:

$$\min \sum_{i \in V_1} \sum_{j \in V_1} c_{ij} y_{ij} + \sum_{j \in V_D} \sum_{l \in V_2} \sum_{m \in V_2} c_{lm} x^j_{lm} \tag{1}$$

subject to:

$$\sum_{i \in V_1} y_{ij} \leq 1, \qquad \forall j \in V_D \tag{2}$$

$$\sum_{i \in V_1} y_{ji} \leq 1, \qquad \forall j \in V_D \tag{3}$$

$$\sum_{i \in V_1} y_{ij} = \sum_{k \in V_1} y_{jk}, \qquad \forall j \in V_D \tag{4}$$

$$\sum_{j \in V_D} y_{j0} = 1, \tag{5}$$

$$\sum_{j \in V_D} y_{0j} = 1, \tag{6}$$

$$\sum_{i \in V'} \sum_{j \in V'} y_{ij} \leq |V'| - 1, \qquad \forall V' \subseteq V_D; |V'| \geq 2 \tag{7}$$

$$x^j_{lm} \leq \sum_{i \in V_1} y_{ij}, \qquad \forall l, m \in V_2; \forall j \in V_D \tag{8}$$

$$\sum_{l \in V_2} \sum_{j \in V_D} x^j_{lm} = 1, \qquad \forall m \in V_C \tag{9}$$

$$\sum_{l \in V_2} x^j_{lm} = \sum_{o \in V_2} x^j_{mo}, \qquad \forall m \in V_C; \forall j \in V_D \tag{10}$$

$$\sum_{l \in V_C} x^j_{jl} = \sum_{o \in V_C} x^j_{oj}, \qquad \forall j \in V_D \tag{11}$$

$$\sum_{l \in V'} \sum_{m \in V'} x^j_{lm} \leq |V'| - \gamma(V'), \forall j \in V_D; \forall V' \subseteq V_C, |V'| \geq 2 \tag{12}$$

$$y_{ij} \in \{0, 1\}, \qquad \forall i, j \in V_1, i \neq j \tag{13}$$

$$x^j_{lm} \in \{0, 1\}, \quad \forall j \in V_D; l, m \in V^j, l \neq m \tag{14}$$

The objective function (1) comprises two terms, the first one represents the length of the first-level trip and the second one the total distance traveled by the truck in the second-level trips. Constraints (2) and (3) state that a trailer point is visited at most once in the first-level trip. Constraints (4) are connectivity constraints for the first-level trip. Constraints (5) and (6) state that the first-level trip departs and ends at the main depot. Constraints (7) are subtour elimination constraints for the first-level trip. Constraints (8) guarantee that second-level trips depart only from trailer points visited in the first-level trip. Constraints (9) state that each customer must be visited exactly once. Constraints (10) guarantee the connectivity of second-level trips. Constraints (11) state that all the second-level trips departing from a trailer point return to it. Constraints (12) prevent subtours in second-level trips, where $\gamma(V')$ is the minimum number of second-level trips needed to serve the demand of the customers in $V' \subseteq V_C$,. Binary variables y_{ij} and x^j_{lm} are defined in (13) and (14) respectively.

3. Metaheuristics for the STTRPSD

Since the STTRPSD generalizes the VRP and the MDVRP, it is clear that only small instances of the STTRPSD could be solved efficiently using the formulation defined by (1)–(14). Therefore, we decided to develop metaheuristics based on greedy randomized adaptive search procedures (GRASP) and evolutionary local search (ELS) to solve it.

3.1. *GRASP/VND*

GRASP [15] is a memory-less multi-start method in which local search is applied to ns initial solutions constructed with a greedy randomized heuristic. Recently, Festa and Resende [16] surveyed the application of GRASP to solve combinatorial optimization problems in various domains. GRASP can be hybridized in different ways, for instance by replacing the local search with another metaheuristic such as tabu search, simulated annealing, variable neighborhood search, iterated local search, among others [36]. Our first metaheuristic for the STTRPSD is a hybrid GRASP/VND in which the local search of GRASP is replaced by a variable neighborhood descent (VND) [21].

3.1.1. *Greedy randomized construction*

Route-first, cluster-second based metaheuristics have shown to be effective solving capacitated node routing problems ranging from the classical VRP [31] to some of its extensions, including the VRP with time windows [23], heterogeneous fleet [33], and pick-up and delivery [39], among others. Following this approach, the greedy randomized construction of the hybrid GRASP/VND is done by means of a tour splitting procedure (hereafter labeled *Split*) that obtains a solution S of the STTRPSD from a giant tour $T = (t_0, t_1, t_2, \ldots, t_j, \ldots, t_n)$ visiting all the customers, where t_j represents the customer in the j-th position and t_0 the main depot.

Giant tours are constructed with a randomized nearest neighbor method with a restricted candidate list (RCL) of size r that ignores capacity constraints and trailer-point selection. Algorithm 1 presents the pseudocode of the randomized nearest neighbor method, note that it is possible to obtain a deterministic nearest neighbor heuristic when $r = 1$, and random permutations of the customers when $r = n$.

Algorithm 1 Randomized nearest neighbor

Parameters: V_c, r
Output: T
Create an empty giant tour T
$t_0 := 0$
$count := 0$
$i := 0$
repeat
    $RCL := \emptyset$
    $sl := min(r, n - count)$
    for $k = 1$ to sl **do**
        $l := \underset{j \in V_c - RCL}{\text{argmin}} \{c_{ij}\}$
        $RCL := RCL \cup \{l\}$
    end for
    Select at random $l^* \in RCL$
    $count := count + 1$
    $t_{count} := l^*$
    $V_c := V_c - \{l^*\}$
    $i := l^*$
until $count = n$
return T

It is worth mentioning that the splitting procedure for the STTRPSD is much more complicated than that of the VRP [31]. Since the distance between trailer points is included in the objective function, the selection and routing of trailer points are important

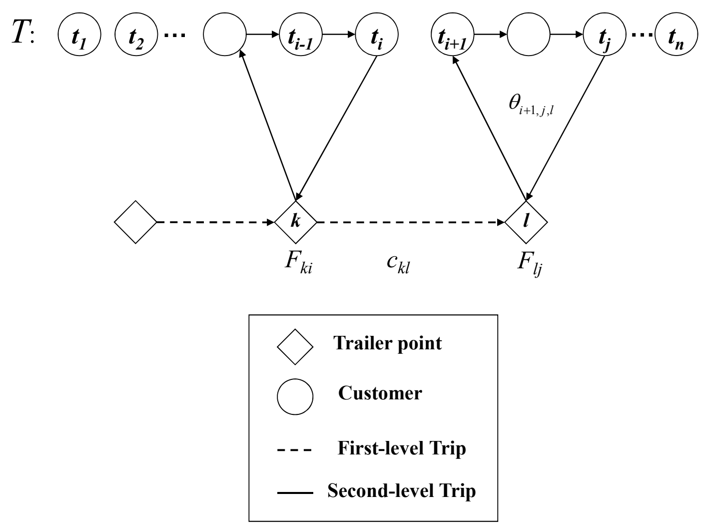


Fig. 2. Principle of the recurrence for the dynamic programming method (*Split*).

decisions. For the STTRPSD, *Split* derives a solution of the STTRPSD from T by optimally splitting it into second-level trips. The selection of open trailer points and the construction of the corresponding first-level trip are also included in the tour splitting procedure.

Split uses a dynamic programming method, in which state $[l, j]$ represents an optimal splitting of $(t_1, t_2, \ldots, t_j)$ with trailer point l for the last trip, and $[0, 0]$ denotes the initial state. Let F_{lj} denote the cost of state $[l, j]$ and θ_{ijk} the cost of a trip visiting customers $(t_i, t_{i+1}, \ldots, t_j)$ from trailer point k. We have the following recurrence relations for any customer t_j and any trailer point l (see Figure 2 for a graphical example).

$$F_{lj} = \begin{cases} 0, & \text{if } l = 0 \text{ and } j = 0 \\ min\left\{F_{ki} + c_{kl} + \theta_{i+1,j,l}\right\}, & i < j : \sum_{u=i+1}^{j} q(t_u) \leq Q_V, k = 0 \text{ if } i = 0 \text{ else } k \in V_D \end{cases} \tag{15}$$

Since states $[l, n]$ $(l \in V_D)$ do not include the return cost to the main depot, the cost of an optimal splitting is $z = min_{l \in V_D}\left\{F_{ln} + c_{l0}\right\}$.

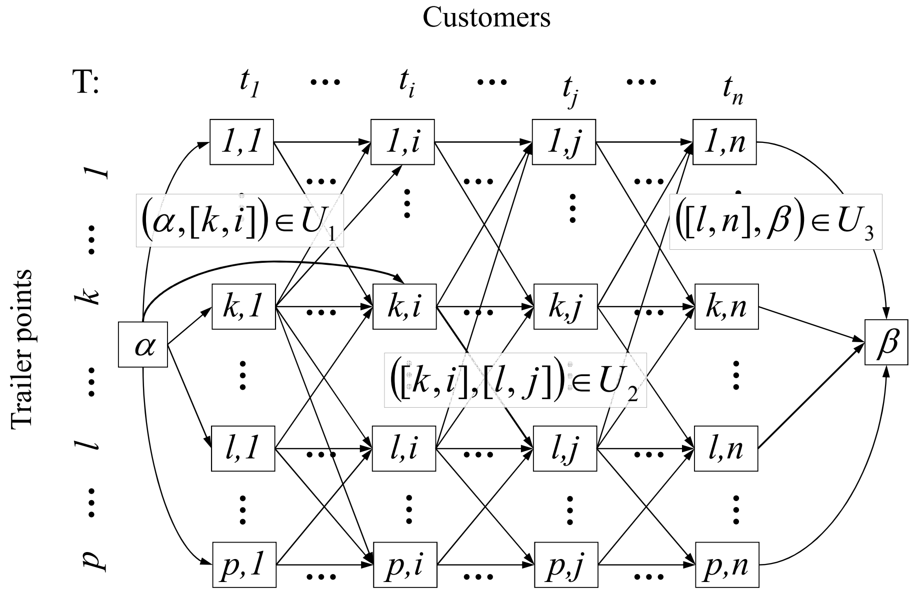


Fig. 3. State Graph $H(X, U, W)$ for the STTRPSD.

The dynamic programming method can be viewed as a shortest path problem in a state graph $H = (X, U, W)$ depicted in Figure 3. The node set X contains $np + 2$ nodes: two copies of the main depot (α and β), acting as source and sink nodes for H, and np nodes for the states $[l, j]$. The arc set U has three types of arcs: U_1 the outgoing arcs of α, U_2 the arcs between internal nodes of the state graph, and U_3 the incoming arcs of β. W is a mapping defining the cost of each arc. Formally, each subset of U and its cost is defined as follows:

$$U_1 = \left\{ (\alpha, [k, i]) : k \in V_D; 1 \le i < n, \sum_{u=1}^{i} q(t_u) \le Q_V \right\} \tag{16}$$

$$w(\alpha, [k, i]) = c(\alpha, k) + c(k, t_1) + \sum_{u=1}^{i-1} c(t_u, t_{u+1}) + c(t_i, k) \tag{17}$$

$$U_2 = \left\{ ([k, i], [l, j]) : k, l \in V_D; 1 \le i < j < n, \sum_{u=i+1}^{j} q(t_u) \le Q_V \right\} \tag{18}$$

$$w(([k, i], [l, j]) = c(k, l) + c(l, t_{i+1}) + \sum_{u=i+1}^{j-1} c(t_u, t_{u+1}) + c(t_j, l) \tag{19}$$

$$U_3 = \left\{ ([l,n],\beta) : l \in V_D \right\} \tag{20}$$

$$w(([l,n],\beta)) = c(l,\beta) \tag{21}$$

Like for the VRP, the shortest path can be computed using Bellman's algorithm for

directed acyclic graphs [12], whose complexity is proportional to the number of arcs. H has n vertical layers of p trailer points each. In the worst case, each node in layer $i = 1, 2, \ldots, n-1$ is linked to all the $p(n-i)$ nodes in subsequent layers. Adding the np outgoing arcs from the source node in the worst case and the p incoming arcs of the sink node, we have:

$$|U| = np + p + \sum_{i=1}^{n-1} p^2(n-i) = p(n+1) + \sum_{i=1}^{n-1} p^2 i = p(n+1) + p^2 \frac{(n)(n-1)}{2} = O(n^2p^2) \tag{22}$$

Even though the number of arcs $|U|$ can be huge, a more precise evaluation shows significant savings. Note that each capacity-feasible subsequence $(t_{i+1}, t_{i+2}, \ldots, t_j)$ of T gives p^2 arcs between vertical layers i and j, because p depots can be chosen for the trip serving customer t_i and p depots for the trip serving customers t_{i+1} to t_j. If the average number of customers per trip is b, the average number of feasible trips is $O(nb)$ and then the number of arcs in U is $O(nbp^2)$. When the average number of clients per trip is small compared to n or, equivalently, if the average customer demand is relatively large compared to vehicle capacity, then $O(nbp^2)$ is significantly smaller than $O(n^2p^2)$. Thus, *Split* has a worst case complexity of $O(nbp^2)$.

Moreover, it is possible to implement the dynamic programming method without generating explicitly the auxiliary graph, using for $F = (F_{lj})$ (see equation (15)) a state table with p rows and $n+1$ columns indexed from 0 to n. The implementation of *Split*, detailed in Algorithm 2, uses two procedures *Develop* and $TourToSol$ (see Appendix B for the details of these procedures). *Develop*(k,i) scans all arcs leaving state $[k,i]$, to update the labels of its successors. Simultaneously the record of the predecessor of each state is stored in two matrices of the same size of F, PD (previous depot) and PC (previous customer). For instance, if the predecessor of state $[l,j]$ is state $[k,i]$ then $PD[l,j] = k$ and $PC[l,j] = i$. After having solved the shortest path problem, the procedure $TourToSol(PD, PC, LD)$ creates a solution of the STTRPSD by backtracking from state $[LD, n]$ using the information of the last satellite depot visited (LD) and the matrices of predecessors.

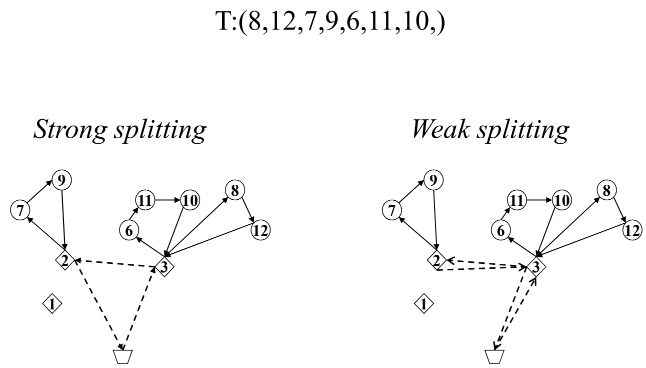


Fig. 4. Example of the difference between strong and weak splitting.

In general, each trailer point may have several trips. Consider a sequence of customers in T close to one trailer point k, with a total demand exceeding Q_V. After splitting, S will contain two consecutive trips departing from k. Such a solution in which each trailer point has consecutive trips is called *strong splitting*. Consider now two capacity-feasible sequences of customers close to k but separated in the giant tour T by other customers. Since the dynamic programming method follows the order of T, after split S will contain two non-consecutive trips departing from trailer point k, so k will be visited twice in the first-level trip; such a solution is called *weak splitting* (Figure 4 gives one example of the two cases). Since the triangle inequality holds the length of the first-level trip can be reduced by making adjacent in T non-consecutive trips with common trailer points (i.e., transforming a weak splitting into a strong one). Then, after the creation of a solution with $TourToSol$, the procedure $EliminateWeakSplitting$ scans the first-level trip to eliminate duplicated visits to the same trailer point. The copy of a trailer point whose elimination produces the greatest reduction in the length of the first-level trip is deleted, and its departing trips reassigned to other visit of the same trailer point. The deletion of copies is repeated until the elimination of the weak splitting.

Figure 5(a) illustrates *Split* on a simple instance of the STTRPSD with $Q_V = 3$, $p = 2$ trailer points (1 and 2), and $n = 4$ customers (3, 4, 5 and 6) with demands 2, 1, 1 and 1, respectively. The length of each square side in the grid is equal to 1 and the distance between nodes is Euclidean. Figure 5(b) shows the results of splitting the giant tour $T = (0, 4, 6, 3, 5)$. Figure 5(c) shows the corresponding state graph, where the arcs of the

Algorithm 2 Split

Input: T

Output: S, solution of the STTRPSD

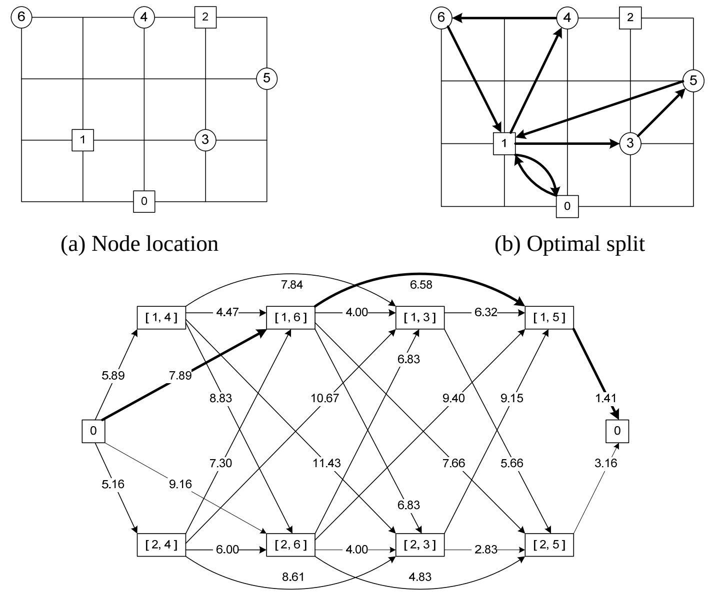
$F[0,0] := 0$
for $k := 1$ to p **do**
  for $i := 0$ to n **do**
    $F[k,i] := \infty$
  end for
end for
$Develop(0,0)$
for $i := 1$ to $n-1$ **do**
  for $k := 1$ to p **do**
    $Develop(k,i)$
  end for
end for
$z := \infty$
for $k := 1$ to p **do**
  if $F[k,n] + c_{k0} < z$ **then**
    $z := F[k,n] + c_{k0}$
    $LD := k$
  end if
end for
$S := TourToSol(PD, PC, LD)$
$S := EliminateWeakSplitting(S)$
return S

Subset	Arc	Trip	Inter-depot distance	Trip distance	Arc Cost
U_1	(0,[1,4])	(1,4,1)	$c_{01} = 1.41$	$\theta_{111} = c_{14} + c_{41} = 4.47$	5.89
U_1	(0,[2,6])	(2,4,6,2)	$c_{02} = 3.16$	$\theta_{122} = c_{24} + c_{46} + c_{62} = 6$	9.16
U_2	([1,4],[1,6])	(1,6,1)	$c_{11} = 0$	$\theta_{221} = c_{16} + c_{61} = 4.47$	4.47
U_2	([1,6],[1,5])	(1,3,5,1)	$c_{11} = 0$	$\theta_{341} = c_{13} + c_{35} + c_{51} = 6.58$	6.58
U_2	([1,6],[2,5])	(2,3,5,2)	$c_{12} = 2.83$	$\theta_{342} = c_{23} + c_{35} + c_{52} = 4.83$	7.66
U_3	([1,5],0)	-	$c_{10} = 1.41$	-	1.41
U_3	([2,5],0)	-	$c_{20} = 3.16$	-	3.16

Table 1

Example of the cost calculation for the 4-customer 2-satellite-depot instance of the STTRPSD.

optimal split are in bold. Table 1 illustrates the calculation of the cost of some arcs of Figure 5(c).



(a) Node location (b) Optimal split

(c) State graph

Fig. 5. *Split* example for the 4-customer sequence $T = (0, 4, 6, 3, 5)$ on a 2-satellite-depot instance of the STTRPSD.

3.1.2. *Variable neighborhood descent*

Solutions obtained from the GRASP construction phase are improved by a VND. VND is a deterministic version of variable neighborhood search [21] in which k_{max} neighborhoods are explored sequentially ($k_{max} \geq 2$). Given the incumbent solution S_0, a subset of the solution space $\mathcal{N}_k(S_0)$ composed of the solutions reachable from S_0 when neighborhood k is applied to it. Algorithm 3, outlines VND with a best-improvement selection strategy in which the best solution of $\mathcal{N}_k(S_0)$ replaces S_0 if it has a smaller cost (line 4). When the incumbent solution is improved the search is reinitialized from the first neighborhood. On the contrary, if the best solution in $\mathcal{N}_k(S_0)$ is not better than S_0 the neighborhood index is increased. VND stops when all the neighborhoods have been explored without improving the incumbent solution (line 11). If a first-improvement selection strategy is used, the search of line 4 is stopped as soon as an improving solution is found (i.e., $\exists S \in \mathcal{N}_k(S_0) : f(S) < f(S_0)$).

Our VND uses two types of neighborhoods : (i) neighborhoods $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ are intended to improve routing within the solution; and (ii) neighborhoods $\mathcal{N}_4$ and $\mathcal{N}_5$ modify the set of open trailer points. Moreover, when applied to second-level trips neighborhoods 1-3 only explore feasible solutions satisfying the capacity constraint. A brief description of the neighborhoods follows:

Algorithm 3 VND with best-improvement selection

Parameters: S_0, neighborhoods structures $\mathcal{N}_1, ..., \mathcal{N}_{k_{max}}$

Output: Improved solution S^*

$flag := true$
$k := 1$
while flag **do**
    $S' := \underset{S \in \mathcal{N}_k(S_0)}{\text{argmin}} \{f(S)\}$
    if $f(S') < f(S_0)$ **then**
        $S_0 := S'$
        $k := 1$
    else
        $k := k + 1$
    end if
    if $k = k_{max} + 1$ **then**
        $flag := false$
        $S^* := S_0$
    end if
end while
return S^*

$\mathcal{N}_1$: Each customer and each trailer point is removed from its current position and reinserted elsewhere. When applied to a customer, the new position could be in the same second-level trip or in another second-level trip; whereas trailer-points changes are made within the first-level trip.

$\mathcal{N}_2$: Two customers or trailer points are exchanged. The two customers may belong to different second-level trips.

$\mathcal{N}_3$: A modified 2-opt in which two edges are exchanged. If the edges belong to the same trip the $\mathcal{N}_3$ reduces to a simple 2-opt for the traveling salesman problem [14]; whereas if the edges belong to a pair of second-level trips with different trailer points, the new trips are assigned to the trailer point from the original trips with the smallest connecting cost. $\mathcal{N}_3$ is inspired by a neighborhood for the capacitated location-routing problem [34].

$\mathcal{N}_4$: Each open trailer point is considered for closure. If the trailer point is closed, its trips are relocated to the remaining trailer points. When relocating a second-level trip, a cheapest insertion of the new trailer point is performed.

$\mathcal{N}_5$: In contrast to $\mathcal{N}_4$, each closed trailer point is considered for opening, only if there exist second-level trips whose cost decreases when relocated to the new trailer point. Again, when relocating a second-level trip, the new trailer point is inserted in the best position.

3.1.3. *Hybrid GRASP/VND for the STTRPSD*

Algorithm 4 outlines the hybrid GRASP/VND for the STTRPSD. In the main cycle, repeated ns times, a giant tour T is constructed with a randomized nearest neighbor heuristic with restricted candidate list of size r. Then, a solution S is produced by applying *Split* to T, and finally S is improved with VND.

Algorithm 4 GRASP/VND for the STTRPSD

Parameters: ns, r
Output: S^*

```
f* := ∞
for i := 1 to ns do
    T := RandomizedNearestNeighbor(V_C, r)
    S := Split(T)
    S := VND(S)
    if f(S) < f* then
        f* = f(S)
        S* := S
    end if
end for
return S*
```

3.2. *Multi-start evolutionary local search*

Evolutionary local search (ELS) [41] can be seen as an evolutionary extension of Iterated Local Search (ILS) [28], in which a single solution is mutated to obtain nc children that are further improved using local search. Following a $(1 + nc)$ selection paradigm, the solution of the next generation is the best among the parent and its nc children [4]. The main loop of mutation and local search is repeated during ni generations.

Although ELS was originally introduced for the solution of optimization problems in telecommunications [41], metaheuristics based on ELS have achieved very good results in different routing problems such as the VRP [32] and the split delivery CARP [5].

We implemented a multi-start variant of ELS, introduced by Prins and Reghioui [35], in which ELS is restarted from nr initial solutions obtained by strongly perturbing the current best solution S^*. By reusing the *Split* and VND procedures described in Sections 3.1.1 and 3.1.2, it is possible to derive the multi-start evolutionary local search for the STTRPSD outlined in Algorithm 5, where the new elements are the *Mutate* and *Concat* operators.

Note that, within the ELS inner loop (lines 12–30 of Algorithm 5), the method alternates between STTRPSD solutions and giant tours. While mutation is applied to giant tours, VND is applied to solutions. When a parent is replaced, a giant tour is derived from it with the procedure (*Concat*) illustrated in Figure 6.

Concat generates a giant tour by creating chains of customers from the second-level trips departing from each open trailer point and concatenating them using the order of the trailer points visited in the first-level trip. As illustrated in Figure 7, the mutation operator (*Mutate*), exchanges b pairs of customers in T. The parameter b is dynamically controlled and varies from 1 to b_{max}. When the parent of ELS is updated the value of b is reset to 1, on the contrary if the parent is not changed the value of b is increased (with a maximum limit b_{max}). We decided to perform the mutation operator in the giant tour because the resulting solution is always feasible and a simple exchange of a pair of customers in T is sometimes enough to perturb the solution structure.

In the proposed multi-start evolutionary local search, the first initial solution is found by splitting the giant tour produced with a deterministic nearest neighbor algorithm ($r = 1$, in Algorithm 1), while the next initial solutions ($i > 1$) are obtained by perturbing and splitting T^* (deduced from S^* by *Concat*). The perturbation is in this case the same mutation operator, but with the exchange of b_{pert} pairs of customers ($b_{pert} \gg b_{max}$).

Algorithm 5 Multi-Start Evolutionary Local Search for the STTRPSD

Parameters: nr, ni, nc, b_{max}, b_{pert}

Output: S^*

```
f* := ∞
b := 1
for i := 1 to nr do
    if i = 1 then
        T := NearestNeighbor(V_C)
    else
        T := Mutate(T*, b_pert)
    end if
    S := Split(T)
    S := VND(S)
    T := Concat(S)
    for j := 1 to ni do
        f̂ := f(S)
        for k := 1 to nc do
            T' := Mutate(T, b)
            S' := Split(T')
            S' := VND(S')
            if f(S') < f̂ then
                f̂ := f(S')
                Ŝ := S'
            end if
        end for
        if f̂ < f(S) then
            S := Ŝ
            T := Concat(S)
            b := 1
        else
            b := min(b + 1, b_max)
        end if
    end for
    if f(S) < f* then
        f* = f(S)
        S* := S
        T* := Concat(S*)
    end if
end for
return S*
```

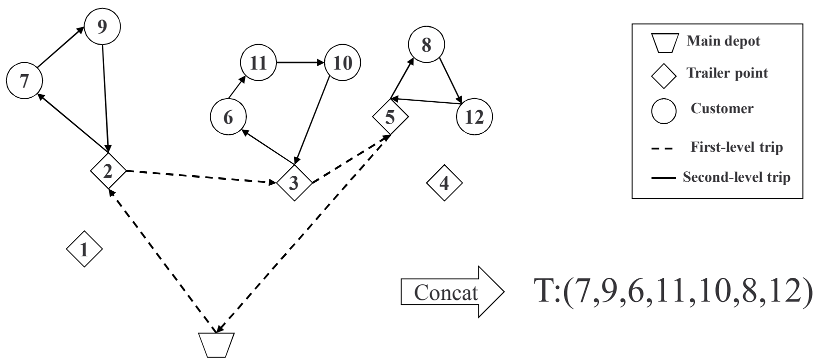


Fig. 6. Principle of procedure *Concat*

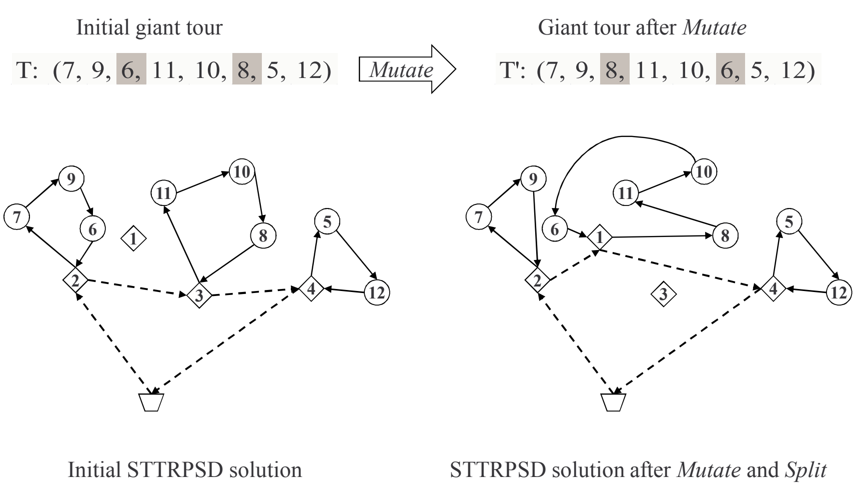


Fig. 7. Principle of procedure *Mutate*

4. Computational study

This section describes the computational experiments performed to evaluate the proposed metaheuristics. Initially, we present the results of several preliminary experiments comparing different options for the building blocks. After the initial fine-tuning process, two test beds were used to evaluate the proposed metaheuristics. The first one is composed of randomly generated instances of the STTRPSD, while the second one corresponds to public instances of the MDVRP.

4.1. *Implementation and STTRPSD test problems*

The proposed methods were tested on a set of 32 randomly generated Euclidean instances with the following characteristics: $n \in \{25, 50, 100, 200\}$, $p \in \{5, 10, 20\}$, and $Q_V \in \{1000, 2000\}$. The coordinates for trailer points and customers are randomly generated in a square grid of size 100×100. There are two problem types based on the distribution of customers and trailer points: clustered (c) and randomly distributed (rd). The customer demands are drawn from a discrete uniform distribution in the interval $[1, 200]$. The data set is available at `http://hdl.handle.net/1992/1125` and Table 2 describes each instance in the testbed. All the metaheuristics were implemented in Java and compiled using Eclipse JDT version 3.3.2. The experiments described in this section were performed on a computer with an Intel Pentium D 945 processor running at 3.4 GHz with 1024 MB of RAM on a Windows XP Professional environment.

4.2. *Preliminary experiments*

We conducted two experiments to analyze the building blocks of the metaheuristics. The first experiment explores the greedy randomized construction procedures. Initial giant tours for the GRASP/VND are obtained using the randomized nearest neighbor (RNN) method using a RCL of size $r = 2$ or $r = 3$. Alternatively, giant tours can be generated through random permutations of customers when $r = n$ (RAND). While generating a solution we can call (or not) the transformation of weak splittings into strong ones (line 21 of Algorithm 2). Combining these options we have a total of six randomized heuristics with the characteristics summarized in Table 3.

Instance	n	p	Q_V	Problem type	Instance	n	p	Q_V	Problem type
1	25	5	1000	c	17	100	10	1000	c
2	25	5	2000	c	18	100	10	2000	c
3	25	5	1000	rd	19	100	10	1000	rd
4	25	5	2000	rd	20	100	10	2000	rd
5	25	10	1000	c	21	100	20	1000	c
6	25	10	2000	c	22	100	20	2000	c
7	25	10	1000	rd	23	100	20	1000	rd
8	25	10	2000	rd	24	100	20	2000	rd
9	50	5	1000	c	25	200	10	1000	c
10	50	5	2000	c	26	200	10	2000	c
11	50	5	1000	rd	27	200	10	1000	rd
12	50	5	2000	rd	28	200	10	2000	rd
13	50	10	1000	c	29	200	20	1000	c
14	50	10	2000	c	30	200	20	2000	c
15	50	10	1000	rd	31	200	20	1000	rd
16	50	10	2000	rd	32	200	20	2000	rd

Table 2

Characteristics of the 32 randomly generated instances.

For each instance in the test set we ran 100 times each randomized heuristic (RH) and computed the average cost and its standard deviation. In Table 3 the column *Avg. Dev.* presents for each method the average deviation with respect to the average cost calculated over all methods. The column *Avg. Mod. CV.* reports the average value of a modified coefficient of variation, in which the standard deviation of the solutions for each method is divided by the average cost of the solutions over all methods. While the former column measures the quality of the solutions, the latter measures their diversity.

Because of the good balance between diversity and solution quality, RH2 ($r = 2$ with strong splitting) was selected as the greedy randomized construction block of GRASP/VND. With $r = 2$ it is possible to obtain better solutions at the expense of just a small loss in diversity, compared to RH3 and RH4 ($r = 3$). On the other hand, RH5 and RH6 (i.e., random permutations) provide a higher diversity, but at the expense of affecting the quality of the solutions. Finally, this experiment shows that the elimination of weak splittings slightly improves the cost of the solutions without affecting much the diversity.

A second experiment, analyzes the contribution of neighborhoods 4 and 5 to the performance of VND. For each instance, we applied two versions of VND to 100 solutions generated with RH2. The first version uses only the routing neighborhoods ($k_{max} = 3$), while the second version includes also neighborhoods 4 and 5 ($k_{max} = 5$). Both options were tested with first-improvement and best-improvement selection criteria. Table 4 re-

Name	Giant Tour	r	Split	Avg. Dev.	Avg. Mod. CV
RH1	RNN	2	Weak	-49.53%	2.88%
RH2	RNN	2	Strong	-50.70%	2.81%
RH3	RNN	3	Weak	-41.87%	3.53%
RH4	RNN	3	Strong	-43.26%	3.50%
RH5	RAND	n	Weak	94.05%	8.67%
RH6	RAND	n	Strong	91.31%	8.81%

Table 3

Preliminary experiments with the greedy randomized construction methods

Name	k_{max}	Option	Avg. Improvement	Avg. Time (ms)
VND1	3	First-improvement	35.0%	182.89
VND2	3	Best-improvement	39.6%	172.68
VND3	5	First-improvement	39.0%	233.76
VND4	5	Best-improvement	42.0%	207.37

Table 4

Preliminary experiments with the VND procedure

ports the average improvement over the initial solution (in %) and the average computation time (in ms) for each method. The use of $k_{max} = 5$ with best-improvement (VND4) produces the best solutions overall. A comparison between the selection criteria shows a marginal (quality) gain between 3 to 4.6% by using the best over the first-improvement strategy; in addition, the best-improvement strategy also proves to be faster than the first-improvement strategy. This experiment also highlights the importance of the selection of trailer points for the quality of the solutions. A comparison of VND2 against VND4 shows that the marginal effect of exploring neighborhoods 4 and 5 is 2.4%. Based on these results we selected VND4 as the local search component of the metaheuristics, even if it takes 20% longer than VND2.

4.3. *Parameter fine-tuning*

To have a fair comparison we allocated a fixed "budget"of 2500 calls of the VND to each metaheuristic to see which one makes the best use of the VND.

For GRASP/VND the selection of the parameters is very simple: $ns = 2500$, and $r = 2$ according to the results of Section 4.2. On the contrary, for the Multi-Start Evolutionary Local Search it is necessary to select the values for b_{max} and b_{pert} and to distribute the 2500 calls of the VND such that $nr \times ni \times nc = 2500$. By setting different values of nc, we implemented two versions of the multi-start evolutionary local search method: 1) a multi-start ELS (henceforth labeled MS-ELS) whit $nc > 1$; and 2) a multi-start ILS (henceforth labeled MS-ILS) with $nc = 1$.

To select the parameters we used a simple (yet effective) fine-tuning procedure: beginning from a promising configuration, one or two parameters were modified, then the current and new configuration were compared using the sign test [9] on a subset of 12 problems. If the new configuration was better we updated the parameters, but if there was no evidence that one configuration was better, the one with the smaller running time was declared the winner and its parameters kept. We stopped the procedure after testing a maximum of 10 configurations. After the fine-tuning procedure the parameters were set to: $nr = 50$, $ni = 10$, $nc = 5$, $b_{max} = 4$ and $b_{pert} = 10$ for the MS-ELS option; and $nr = 125$, $ni = 20$, $nc = 1$, $b_{max} = 4$ and $b_{pert} = 10$ for the MS-ILS option.

4.4. *Results*

We used three heuristics as benchmarks to compare the performance of the metaheuristics. The first one is a simple cluster-first, route-second (henceforth labeled CFRS) approach in which each customer is assigned to its nearest trailer point; then the customers assigned to each open trailer point are routed using an insertion heuristic (i.e., a VRP is solved for each open trailer point); and finally, the first-level trip that visits the set of open trailer points is derived from the same insertion heuristic. The second heuristic enhances the solutions constructed with cluster-first, route-second by using VND4 (CFRS+VND). The third and last heuristic, called iterated route-first, cluster-second (henceforth labeled IRFCS) repeats RH2 2500 times.

Tables 5 and 6 presents the results of the three benchmark heuristics and the proposed metaheuristics on the set of 32 randomly generated instances, respectively. Additionally, Table 6 includes the results of a hybrid GRASP×ELS proposed in [40]. In both tables we present the best, worst and average results over ten runs for all the randomized methods; the last two rows report the number of times each method found the best-known solution (NBKS) and the average deviation (in %) above the best-known solution (BKS). The last column reports the cost of the best-known solution for each problem, and every BKS is highlighted in bold type when found by a given method. Table 7 shows the average running times (in minutes) for each method over all instances. We do not report exact running times for CFRS and CFRS+VND since they take on average less than one second.

Even though the CFRS heuristic is extremely fast, its average deviation above the best

Instance	n	p	Type	CFRS	CFRS+VND	IRFCS Best	IRFCS Worst	IRFCS Average	BKS
1	25	5	c	444.08	**405.46**	420.34	435.62	427.48	405.46
2	25	5	c	444.08	391.62	390.60	407.58	398.56	374.79
3	25	5	rd	696.73	585.96	596.17	644.50	618.61	584.03
4	25	5	rd	640.01	526.27	530.48	568.88	548.32	508.48
5	25	10	c	460.28	**386.45**	398.41	409.12	404.86	386.45
6	25	10	c	460.28	386.45	391.43	406.42	401.46	380.86
7	25	10	rd	789.70	582.64	597.13	628.87	613.02	573.96
8	25	10	rd	789.70	582.64	521.67	559.78	542.26	506.37
9	50	5	c	625.67	583.41	641.15	655.86	646.42	583.07
10	50	5	c	574.17	560.17	594.72	616.69	608.57	516.98
11	50	5	rd	1177.25	**870.51**	994.62	1031.36	1012.40	870.51
12	50	5	rd	980.57	787.79	895.60	935.34	907.91	766.03
13	50	10	c	471.43	**387.83**	424.53	438.86	433.50	387.83
14	50	10	c	460.36	381.32	415.45	426.20	421.57	367.01
15	50	10	rd	1034.77	847.49	892.42	953.81	931.37	811.28
16	50	10	rd	1013.20	758.95	855.75	893.63	874.40	731.53
17	100	10	c	705.19	640.01	724.13	755.20	742.97	614.02
18	100	10	c	665.76	555.31	679.70	702.49	691.22	547.44
19	100	10	rd	1544.01	1416.60	1569.15	1612.74	1593.06	1275.76
20	100	10	rd	1290.79	1167.97	1378.20	1440.09	1408.73	1097.28
21	100	20	c	820.00	668.04	768.01	789.28	781.21	642.61
22	100	20	c	808.61	643.16	692.03	727.02	714.43	581.56
23	100	20	rd	1392.01	1192.83	1374.35	1435.41	1410.23	1143.10
24	100	20	rd	1342.10	1138.84	1321.92	1373.19	1348.44	1060.75
25	200	10	c	1004.80	849.63	1032.04	1068.24	1049.39	822.52
26	200	10	c	878.59	734.63	936.67	961.79	949.52	714.33
27	200	10	rd	2391.08	2026.04	2305.69	2344.00	2322.63	1761.10
28	200	10	rd	1951.77	1515.01	2028.96	2081.73	2050.88	1445.94
29	200	20	c	1098.70	950.21	1134.92	1169.78	1152.02	909.46
30	200	20	c	1036.27	862.35	1040.35	1069.29	1056.52	815.51
31	200	20	rd	2251.76	1691.43	2142.39	2191.40	2167.92	1614.18
32	200	20	rd	2019.44	1559.43	1956.08	2004.72	1975.18	1413.32
NBKS				0	4	0	-	-	
Avg. Deviation Above BKS(%)				26.17	4.70	17.45	21.98	19.81	

Table 5

Results of the heuristics on the 32-instance testbed

solutions is large (26.17%) compared to the average deviation of CFRS+VND (4.70%) which also runs in less than one second. IRFCS produces, on average, solutions of better quality (21.98%) than CFRS and still with short running times. However, CFRS frequently outperforms IRFCS in the clustered problems. Also, the results of IRFCS have a large variability, with a gap of 4.53% between its best and worst results.

As can be seen in Table 6, the metaheuristics proposed in this chapter outperform the

				GRASP × ELS			GRASP/VND			MS-ELS			MS-ILS			
Instance	n	p	Type	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average	Best	Worst	Average	BKS
1	25	5	c	**405.46**	405.46	405.46	**405.46**	405.46	405.46	**405.46**	405.46	405.46	**405.46**	405.46	405.46	405.46
2	25	5	c	**374.79**	374.79	374.79	**374.79**	374.79	374.79	**374.79**	374.79	374.79	**374.79**	374.79	374.79	374.79
3	25	5	rd	**584.03**	584.03	584.03	**584.03**	584.03	584.03	**584.03**	584.03	584.03	**584.03**	584.03	584.03	584.03
4	25	5	rd	**508.48**	508.48	508.48	**508.48**	508.48	508.48	**508.48**	508.48	508.48	**508.48**	508.48	508.48	508.48
5	25	10	c	**386.45**	386.45	386.45	**386.45**	386.45	386.45	**386.45**	386.45	386.45	**386.45**	386.45	386.45	386.45
6	25	10	c	**380.86**	380.86	380.86	**380.86**	380.86	380.86	**380.86**	380.86	380.86	**380.86**	380.86	380.86	380.86
7	25	10	rd	**573.96**	573.96	573.96	**573.96**	573.96	573.96	**573.96**	573.96	573.96	**573.96**	573.96	573.96	573.96
8	25	10	rd	**506.37**	506.37	506.37	**506.37**	506.37	506.37	**506.37**	506.37	506.37	**506.37**	506.37	506.37	506.37
9	50	5	c	**583.07**	585.50	585.50	**583.07**	583.07	583.07	**583.07**	583.41	583.10	**583.07**	583.07	583.07	583.07
10	50	5	c	**516.98**	516.98	516.98	**516.98**	516.98	516.98	**516.98**	516.98	516.98	**516.98**	516.98	516.98	516.98
11	50	5	rd	**870.51**	870.51	870.51	**870.51**	870.51	870.51	**870.51**	870.51	870.51	**870.51**	870.51	870.51	870.51
12	50	5	rd	**766.03**	766.03	766.03	**766.03**	766.03	766.03	**766.03**	766.03	766.03	**766.03**	766.03	766.03	766.03
13	50	10	c	389.07	389.76	389.76	**387.83**	387.83	387.83	**387.83**	387.83	387.83	**387.83**	387.83	387.83	387.83
14	50	10	c	**367.01**	367.01	367.01	**367.01**	367.01	367.01	**367.01**	367.01	367.01	**367.01**	367.01	367.01	367.01
15	50	10	rd	**811.28**	811.28	811.28	**811.28**	811.28	811.28	**811.28**	811.28	811.28	**811.28**	811.28	811.28	811.28
16	50	10	rd	**731.53**	735.66	735.66	**731.53**	731.53	731.53	**731.53**	731.53	731.53	**731.53**	731.53	731.53	731.53
17	100	10	c	615.00	618.29	618.29	**614.02**	614.61	614.22	614.20	614.31	614.30	**614.02**	615.32	614.41	614.02
18	100	10	c	548.36	550.08	550.08	**547.44**	547.44	547.44	**547.44**	548.11	547.64	**547.44**	548.11	547.57	547.44
19	100	10	rd	1284.48	1297.80	1297.80	1278.90	1290.52	1285.45	**1275.76**	1285.38	1280.65	1280.02	1286.14	1282.79	1275.76
20	100	10	rd	1099.22	1114.45	1114.45	**1097.28**	1107.37	1102.39	**1097.28**	1097.28	1097.28	**1097.28**	1103.43	1097.90	1097.28
21	100	20	c	**642.61**	646.00	646.00	**642.61**	643.93	643.12	**642.61**	643.93	642.79	**642.61**	642.61	642.61	642.61
22	100	20	c	**581.56**	585.07	585.07	**581.56**	581.56	581.56	**581.56**	583.71	581.78	**581.56**	583.71	582.18	581.56
23	100	20	rd	1148.79	1165.28	1165.28	1146.58	1151.17	1148.80	**1143.10**	1151.29	1147.11	**1143.10**	1150.34	1146.66	1143.10
24	100	20	rd	1063.85	1077.30	1077.30	1068.25	1076.44	1071.18	**1060.75**	1066.54	1064.04	**1060.75**	1064.99	1062.84	1060.75
25	200	10	c	835.91	851.10	851.10	829.89	838.37	833.45	827.10	836.39	830.99	**822.52**	835.02	828.37	822.52
26	200	10	c	728.68	740.14	740.14	717.46	725.16	722.42	715.37	729.98	723.17	**714.33**	725.99	719.99	714.33
27	200	10	rd	1816.64	1845.04	1845.04	1787.24	1816.84	1808.32	**1761.10**	1807.95	1787.63	1763.30	1805.70	1783.24	1761.10
28	200	10	rd	1471.65	1512.90	1512.90	1469.35	1486.87	1480.18	1454.90	1475.95	1461.06	**1445.94**	1470.70	1458.15	1445.94
29	200	20	c	923.55	935.17	935.17	918.29	922.75	920.24	912.87	920.76	916.17	**909.46**	923.22	913.51	909.46
30	200	20	c	820.99	838.58	838.58	819.28	822.15	820.94	**815.51**	824.51	820.64	820.67	824.84	822.19	815.51
31	200	20	rd	1660.42	1675.39	1675.39	1637.01	1662.91	1655.92	1620.47	1648.79	1632.10	**1614.18**	1649.12	1631.61	1614.18
32	200	20	rd	1444.70	1479.96	1479.96	1423.72	1449.94	1442.01	1420.45	1449.77	1432.01	**1413.32**	1438.97	1424.77	1413.32
NBKS				17			21			25			29			
Avg. Deviation Above BKS(%)				0.57	1.29	0.91	0.29	0.68	0.52	0.08	0.58	0.31	0.03	0.54	0.25	

Table 6. Results of the metaheuristics on the 32-instance testbed

			Average running time (min)				
Instance	n	p	IRFCS	GRASP × ELS	GRASP/VND	MS-ELS	MS-ILS
1	25	5	0.01	0.24	0.20	0.17	0.19
2	25	5	0.01	0.23	0.21	0.18	0.19
3	25	5	0.01	0.25	0.21	0.19	0.21
4	25	5	0.01	0.23	0.18	0.16	0.17
5	25	10	0.02	0.29	0.22	0.20	0.22
6	25	10	0.02	0.27	0.25	0.19	0.21
7	25	10	0.02	0.28	0.25	0.21	0.23
8	25	10	0.02	0.25	0.21	0.19	0.21
9	50	5	0.03	1.73	1.23	0.94	1.18
10	50	5	0.04	1.70	1.20	0.78	0.95
11	50	5	0.04	1.48	1.29	0.85	1.02
12	50	5	0.04	1.52	1.24	0.83	0.98
13	50	10	0.05	1.95	1.23	1.16	1.37
14	50	10	0.07	1.95	1.27	1.02	1.25
15	50	10	0.05	1.64	1.39	0.99	1.20
16	50	10	0.07	1.54	1.29	0.86	1.05
17	100	10	0.14	12.61	8.48	4.45	5.80
18	100	10	0.19	13.21	8.72	5.09	6.49
19	100	10	0.15	9.14	9.70	3.82	4.96
20	100	10	0.20	9.47	9.06	3.54	4.34
21	100	20	0.29	12.38	9.45	4.03	4.82
22	100	20	0.49	13.73	9.35	5.08	6.00
23	100	20	0.29	10.03	10.43	4.09	5.02
24	100	20	0.48	10.61	9.89	4.11	4.91
25	200	10	0.49	62.65	60.86	15.44	19.24
26	200	10	0.60	73.34	60.95	15.39	19.47
27	200	10	0.47	52.23	63.92	13.90	17.19
28	200	10	0.58	58.13	64.33	13.02	15.48
29	200	20	0.78	71.43	64.70	18.09	22.91
30	200	20	1.19	80.67	62.91	19.55	24.41
31	200	20	0.78	57.27	68.20	16.39	20.47
32	200	20	1.19	62.95	68.59	16.18	20.14
Average			0.28	19.54	18.79	5.35	6.63

Table 7

Average times (over 10 runs) of the proposed methods on the set of randomly generated instances

benchmark heuristics and the previous hybrid GRASP×ELS reported in [40]. Among the proposed metaheuristics, multi-start evolutionary local search is both faster and more accurate than GRASP/VND. The MS-ILS version found 29 out of 32 best known solutions and was 2.8 times faster than GRASP/VND; while the MS-ELS version was 3.5 times faster, and found 25 out of 32 best known solutions. Also important is the fact that multi-start evolutionary local search reports very small average gaps with respect to best known solutions. While MS-ILS and MS-ELS achieved average gaps as small as 0.25% and 0.31%, respectively; the GRASP/VND and GRASP×ELS report larger average gaps of 0.52% and 0.91%, respectively. Despite of the slight differences, remarkably these experiments highlight the robustness of the proposed metaheuristics: none of them has deviations to BKS greater than 1.0% even under their worst performance.

The comparison of GRASP/VND against IRFCS shows the effect of the VND on the quality of the solutions at the expense of computational effort. As can be seen in Table 7, VND represents more than 98% of the running time of GRASP/VND. On the contrary, MS-ELS and MS-ILS exploit better the local search. First and foremost, they scale much better than GRASP/VND as the number of customers increases (Figure 8). Second, as shown in Figure 9, they have better Time-to-Target distributions [1] than GRASP/VND. Figure 9 was constructed by applying each metaheuristic 200 times to a fixed instance and recording the time needed to find a solution with objective function at least as good as a given target value. By analyzing the Time-to-Target distributions, GRASP/VND needs approximately 150 seconds to have a probability of 90% to find a solution as good as the given target value; on the other hand, MS-ILS needs just 50 seconds and MS-ELS only 60 seconds to reach the same quality.

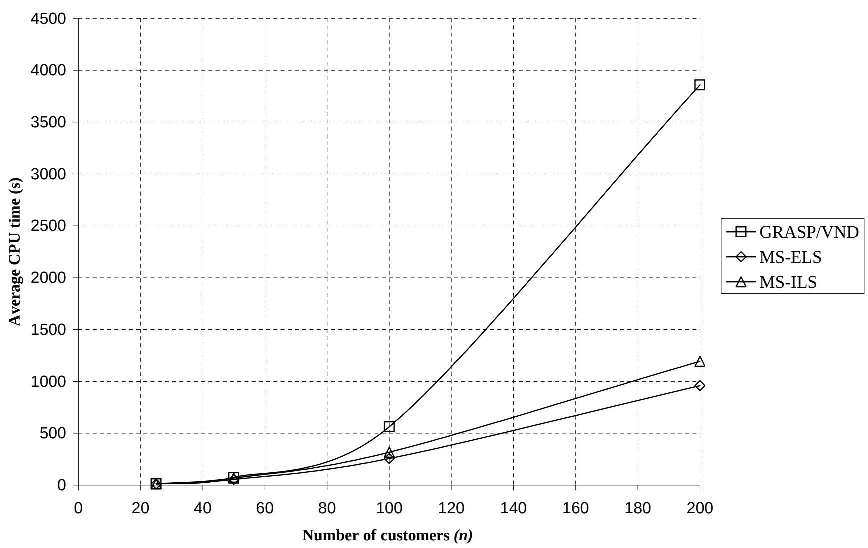


Fig. 8. Average running times for each metaheuristic.

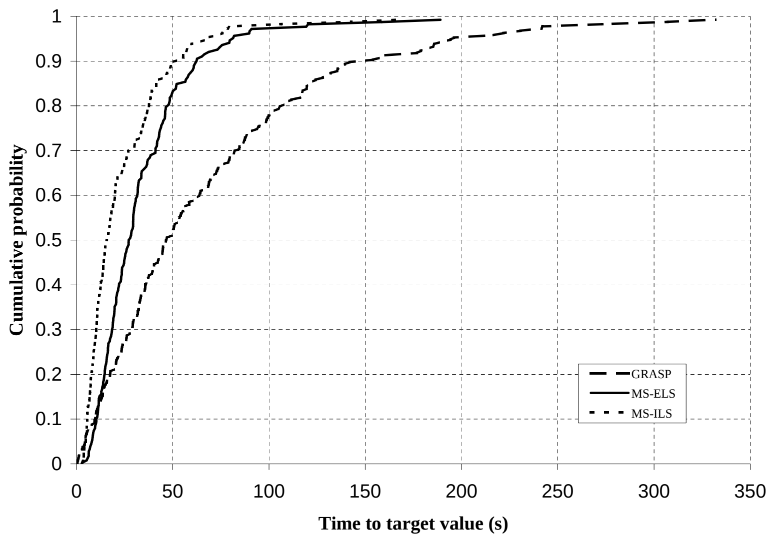


Fig. 9. Time-to-target plot for a STTRPSD with $n = 100$ and $p = 20$

4.5. *Multi-depot VRP*

Since the STTRPSD is a new problem, no published solution method is available for comparison. Because the MDVRP is a special case of the STTRPSD where $c_{ij} = 0$, $\forall i, j \in V_1$, we tested the proposed methods on the MDVRP, and compare them against the best published metaheuristics designed to solve it.

Without changing their parameters GRASP/VND, MS-ELS and MS-ILS were applied 10 times for each instance in the set of test problems commonly used for the MDVRP [11] (available at `http://neumann.hec.ca/chairedistributique/data/mdvrp/`). Given that the STTRPSD does not have route-length constraints, we only solve the 11 MDVRP instances without such constraints. Table 8 presents the results of this experiment. The best known solutions taken from [30] were used to calculate the average deviations given in the last row of Table 8 and best average results for each instance are in bold. As a reference, the results of Pisinger and Ropke ALNS [30] and Cordeau et al. tabu search [11] are included in Table 8.

Although the methods presented in this work are not tailored to solve the MDVRP, the proposed multi-start evolutionary local search performs well and stands as a competitive alternative. Overall, ALNS is the best method, with the smallest deviation (0.40%) and the best average performance on 6 out of the 11 instances, closely followed by MS-ILS and tabu search with the best average performance in 4 out of 11 instances. Remarkably, MS-ILS achieved an average deviation of 0.49% against 0.50% of tabu search. Not far along is MS-ELS which is also effective in the solution of the MDVRP with an average deviation of 0.61%. Finally, GRASP/VND does not perform as well in the MDVRP: despite the fact that the best results have an average deviation of 4.96%, its average deviation reaches 9.98%. In terms of computing times, the methods presented in this chapter are slower than ALNS and tabu search, specially on problems P18 and P21 with over 200 customers. Nevertheless, our algorithms could be easily accelerated for the MDVRP, for instance, by implementing the Split procedure in $O(nbp)$ instead of the $O(nbp^2)$ of equation (22). Because the target of this chapter is the STTRPSD and not the MDVRP, we decided not to change our code to tailor it to the MDVRP.

Instance				ALNS [a]			CGL [b]			GRASP/VND			MS-ELS			MS-ILS		
	n	p	Best known	Cost [c]	Best [d]	T [e]	Cost [f]	Best [g]	T [h]	Cost [i]	Best [j]	T [k]	Cost [i]	Best [j]	T [k]	Cost [i]	Best [j]	T [k]
P01	50	4	576.87	**576.87**	576.87	0.48	**576.87**	576.87	3.24	610.79	592.21	1.26	**576.87**	576.87	0.77	**576.87**	576.87	0.89
P02	50	4	473.53	**473.53**	473.53	0.47	473.87	473.53	3.46	556.35	529.64	1.40	**473.53**	473.53	1.01	**473.53**	473.53	1.12
P03	75	5	641.19	**641.19**	641.19	1.07	645.15	641.19	5.66	694.08	648.68	3.92	643.50	641.19	1.79	643.44	641.19	2.09
P04	100	2	1001.04	**1006.09**	1001.04	1.47	1006.66	1001.59	7.79	1071.49	1055.26	8.02	1014.07	1005.67	2.86	1008.10	1003.62	3.50
P05	100	2	750.03	752.34	751.26	2.00	753.34	750.03	8.21	803.93	769.37	8.66	**751.49**	751.15	2.63	752.54	751.15	3.13
P06	100	3	876.5	883.01	876.7	1.55	**877.84**	876.5	7.65	963.10	924.68	8.16	883.73	878.90	2.95	884.42	880.69	3.51
P07	100	4	881.97	**889.36**	881.97	1.47	891.95	885.8	7.71	955.76	925.80	8.70	893.05	887.16	2.97	892.59	888.65	3.55
P12	80	2	1318.95	1319.13	1318.95	1.25	**1318.95**	1318.95	5.57	1409.02	1326.85	3.69	**1318.95**	1318.95	1.92	**1318.95**	1318.95	2.20
P15	160	4	2505.42	2519.64	2505.42	4.22	2534.13	2505.42	14.06	2809.46	2553.80	28.65	2521.75	2505.42	7.83	**2508.05**	2505.42	9.30
P18	240	6	3702.85	3736.53	3702.85	6.98	**3710.49**	3702.85	24.85	4075.48	4209.56	96.84	3727.80	3702.85	18.81	3737.64	3702.85	22.72
P21	360	9	5474.84	**5501.58**	5474.84	9.70	5535.99	5474.84	48.16	6187.00	5903.63	325.23	5555.78	5504.04	44.35	5522.02	5490.55	52.96
Average Time						2.79			12.40			44.96			7.99			9.54
Avg. Deviation above BKS (%)				0.40	0.02		0.50	0.04		9.98	4.96		0.61	0.18		0.49	0.18	

Table 8. Results for the MDVRP

[a] Results of the best variant of Pisinger and Ropke ALNS [30] with 50000 iterations
[b] Results of Cordeau et al. tabu search [11]
[c] Average cost over 10 runs
[d] Best solution found over 10 runs
[e] Average time in minutes on a 3GHz Pentium 4 over 10 runs
[f] Cost of the solution found with standard parameters settings in a single run
[g] Best solution found during sensitivity analysis with different parameter settings
[h] Time in minutes on a Sun Sparcstation 10
[i] Average cost over 10 runs
[j] Best solution found over 10 runs
[k] Average time in minutes on a 3.4GHz Pentium D over 10 runs

5. Conclusion and future work

In this chapter, we introduced the STTRPSD, a generalization of the VRP arising from practical applications such as milk collection and postal services. To solve the STTRPSD we proposed two metaheuristics: a hybrid GRASP/VND and a multi-start evolutionary local search. Both of them perform very well when compared against cluster-first route-second, iterated route-first cluster-second and VND heuristics. The results of the computational experiments on a set of 32 randomly generated instances also unveil the robustness of the proposed metaheuristics, all of them achieving gaps to best known solutions of less than 1% even in the worst case. Among the proposed methods, the multi-start evolutionary local search is more accurate, faster, and scales better (as the number of customers increases) than the GRASP/VND. Finally, when tested on the MDVRP, a special case of the STTRPSD, the proposed multi-start evolutionary local search obtains results that are competitive to those achieved by the state-of-the art ALNS [30] and tabu search [11]. Future research directions include the derivation of a lower bound to evaluate the quality of solutions found with heuristics and metaheuristics, the development of population metaheuristics, and the extension of the methods to the case with multiple vehicles.

Appendix

Appendix A. Nomenclature

A.1. *Notation for the integer programming formulation*

G: Graph for the formulation of the STTRPSD, $G = (V, A)$.
V: Set of nodes of G.
V_C: Subset of V representing the customers.
V_D: Subset of V representing the satellite depots.
V_1: Subset of V representing the nodes that can be visited in the first-level trip.
V_2: Subset of V representing the nodes that can be visited in second-level trips.
V^j: Subset of V representing the nodes that can be visited in second-level trips departing from satellite depot j.
A: Set of arcs of G.
c_{ij}: Cost (length) of the arc between nodes i and j ($i, j \in V$).
n: Number of customers.
p: Number of satellite depots.
q_i: Demand of customer i ($i \in V_C$).
Q_V: Capacity of the truck.
Q_T: Capacity of the trailer.
$\gamma(V')$: Minimum number of second-level trips needed to serve the demand of the customers in $V' \subseteq V_C$, $\gamma(V') = \lceil \frac{\sum_{i \in V'} q_i}{Q_V} \rceil$
x^j_{lm}: Binary variable, $x^j_{lm} = 1$ if arc (l, m) is traversed by the truck in a second-level trip departing from satellite depot j ($l, m \in V^j$); $x^j_{lm} = 0$ otherwise.
y_{ij}: Binary variable, $y_{ij} = 1$ if arc (i, j) is traversed in the first-level trip ($i, j \in V_1$); $y_{ij} = 0$ otherwise.

A.2. *Notation used in the metaheuristics*

A.2.1. *General notation*

S: A solution of the STTRPSD.
$f(S)$: Objective function of solution S calculated using equation (1).
S^*: Best solution found during the execution of a method.
f^*: Cost of the best solution.

A.2.2. *GRASP*

ns: Number of iterations.

r: Cardinality of the restricted candidate list.

A.2.3. *Split*

T: Giant tour visiting all the customers in V_C, $T = (t_0, t_1..., t_n)$.

t_j: Customer in position j of giant tour T.

F_{lj}: Cost of state $[l, j]$ in the dynamic programming method.

θ_{ijk}: Cost of the second-level trip $(k, t_i, \ldots, t_j, k)$, visiting the customers from position i to position j of giant tour T from satellite depot k.

H: Auxiliary graph, $H = (X, U, W)$.

X: Nodes of H, representing the states of the dynamic programming method.

U: Arcs of H.

W: Mapping defining the cost of the arcs of H.

PD : Matrix with the record of the preceding satellite depot in the dynamic programming method.

PD : Matrix with the record of the preceding customer in the dynamic programming method.

LD : Last satellite depot used in the solution of the shortest path problem in H.

A.2.4. *VND*

$\mathcal{N}_k(S_0)$: Subset of the solution space composed of the solutions reachable from solution S_0 when neighborhood k is applied to it.

k_{max}: Number of neighborhoods.

A.2.5. *Multi-start evolutionary local search*

nr: Number of restarts.

ni: Number of iterations of ELS.

nc: Number of children obtained by mutation in each iteration.

b_{max}: Maximum number of pairs for the mutation operator.

b_{pert}: Number of pairs for the perturbation of the best tour in each restart.

T^*: Giant tour of the best solution.

T': Giant tour obtained with the mutation operator.

S': Child generated after mutation and tour splitting.

$\hat{S}$: Best child generated in each iteration.

$\hat{f}$: Cost of the best child.

Appendix B. Procedures used in *Split*

Procedure *Develop*(k, i) scans the successors of state $[k, i]$ to update their labels and the predecessor records PD and PC. It uses a function, $MaxRank(i)$ that returns the largest index j such that the second-level trip $(L, t_i..., t_j, L)$ is feasible for any $L \in V_D$.

Algorithm 6 *Develop*$[k, i]$

Input: T, k, i

```
last := MaxRank(i + 1)
cost := F(k, i)
w := t_{i+1}
u := w
for j := i to last do
    v := t_j
    cost := cost + c_{uv}
    u := v
    for L := 1 to p do
        Fnew := cost + c_{kl} + c_{Lw} + c_{vL}
        if Fnew ≤ F[L, j] then
            F[L, j] := Fnew
            PD[L, j] := k
            PC[L, j] := i
        end if
    end for
end for
```

After solving the shortest path problem in the auxiliary graph H using Algorithm 2, the solution S associated with a giant tour T can be deduced by backtracking from state $[LD, n]$. This is done with procedure $TourToSol$ using the information stored in LD, PD and PC.

Algorithm 7 $TourToSol(LD, PD, PC)$

Input: LD, PD, PC

Output: S, solution of the STTRPSD

Create an empty solution of the STTRPSD S
Create flt an empty first-level trip departing from the main depot
$L := LD$
$lisd := L$
Insert L at the beginning of flt
$j := n$
repeat
  $k := PD[L, j]$
  $i := PC[L, j]$
  if $lisd \neq L$ **then**
    Insert L at the beginning of flt
    $lisd := L$
  end if
  Create a second-level trip $slt := (L, t_{i+1}, \ldots, t_j, L)$
  add slt to S
  $L := k$
  $j := i$
until $j := 0$
add flt to S
return S

References

[1] R. Aiex, M. Resende, and C. Ribeiro. TTTplots: A Perl program to create time-to-target plots. *Optimization Letters*, 1:335–366, 2007.

[2] R. Baldacci and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming Ser. A*, 120:347–380, 2009.

[3] R. Baldacci, P. Toth, and D. Vigo. Recent advances in vehicle routing exact algorithms. *4OR - A Quarterly Journal of Operations Research*, 5:269–298, 2007.

[4] T. Bäck, F. Hoffmeister, and H.-P. Schwefel. A survey of evolution strategies. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 2–9, San Diego, 1991. Morgan Kaufmann.

[5] J.-M. Belenguer, E. Benavent, N. Labadi, C. Prins, and M. Reghioui. Split delivery capacitated arc routing problem: lower bound and metaheuristic. *Transportation Science*, to appear.

[6] M. Caramia and F. Guerriero. A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society*, DOI:10.1057/jors.2009.59, 2009.

[7] I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29:33–51, 2002.

[8] G. Claassen and T. Hendriks. An application of special ordered sets to a periodic milk collection problem. *European Journal of Operational Research*, 180:754–769, 2007.

[9] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, 1998.

[10] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis. VRP with Time Windows. In P. Toth and D. Vigo, editors, *The vehicle routing problem*, pages 157 – 193. SIAM, 2002.

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

[12] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*. The MIT Press, Cambridge, 2001.

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

[14] G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6:791–812, 1958.

[15] T. Feo and M. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[16] P. Festa and M. Resende. An annotated bibliography of GRASP, Part II: Applications. *International Transactions in Operational Research*, 16:131–172, 2009.

[17] M. Gendreau, J.-Y. Potvin, O. Bräysy, G. Hasle, and A. Løkketangen. Metaheuristics for the Vehicle Routing Problem and Its Extensions: A Categorized Bibliography. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 143–170. Springer, 2008.

[18] G. Ghiani, G. Improta, and G. Laporte. The capacitated arc routing problem with intermediate facilities. *Networks*, 37:134–143, 2001.

[19] B. Golden, S. Raghavan, and E. Wasil. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, 2008.

[20] J. Gonzalez-Feliu, G. Perboli, R. Tadei, and D. Vigo. The two-echelon capacitated vehicle routing problem. Technical report, Control and Computer Engineering Department, Politecnico di Torino, 2007.

[21] P. Hansen and N. Mladenovic. Variable neighbourhood search: Principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

[22] A. Hoff and A. Løkketangen. A tabu search approach for milk collection in western Norway using trucks and trailers. In *TRISTAN VI: The Sixth Triennial Symposium on Transportation Analysis*, Phuket, Thailand, June 10–15, 2007.

[23] N. Labadi, C. Prins, and M. Reghioui. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO Operations Research*, 42:415–431, 2008.

[24] G. Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics*, 54:811 – 819, 2007.

[25] G. Laporte and F. Semet. Classical Heuristics for the Capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 109–128. SIAM, 2002.

[26] L. Levy and L. Bodin. Scheduling of Local Delivery Carriers for the United States Postal Service. In M. Dror, editor, *Arc Routing: Theory, Solutions and Applications*, pages 419–442. Kluwer, 2000.

[27] S.-W. Lin, V. F. Yu, and S.-Y. Chou. Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, 36:1683–1692, 2009.

[28] H. Lourenço, O. Martin, and T. Stützle. Iterated Local Search. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 321-353. Kluwer, 2003.

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

[30] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34:2403–2435, 2007.

[31] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31:1985–2002, 2004.

[32] C. Prins. A GRASP × Evolutionary Local Search Hybrid for the Vehicle Routing Problem. In F. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, pages 35–53. Springer, 2009.

[33] C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22:916–928, 2009.

[34] C. Prins, C. Prodhon, and R. Wolfler-Calvo. 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, 2006.

[35] C. Prins and M. Reghioui. A hybrid GRASP×evolutionary local search for the split delivery vehicle routing problem. In *Odysseus 2009*, Cezme, Izmir, Turkey, May 26-29 2009.

[36] M.G.C. Resende. Metaheuristic Hybridization with Greedy Randomized Adaptive Search Procedures. In Z-L. Chen and S. Raghavan, editors, *TutORials in Operations Research*, pages 295–319. INFORMS, 2008.

[37] S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33:894–909, 2006.

[38] P. Toth and D. Vigo. *The Vehicle Routing Problem.* SIAM, Philadelphia, 2002.

[39] N. Velasco, P. Castagliola, P. Dejax, C. Guéret, and C. Prins. A Memetic Algorithm for a Pick-Up and Delivery Problem by Helicopter, In F. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, pages 173–190. Springer, 2009.

[40] J.G. Villegas, A.L. Medaglia, C. Prins, C. Prodhon, and N. Velasco. GRASP/evolutionary local search hybrids for a truck and trailer routing problem. In *MIC 2009: The VIII Metaheuristics International Conference*, Hamburg, July 13–16, 2009.

[41] S. Wolf and P. Merz. Evolutionary Local Search for the Super-peer Selection Problem and the p-hub Median Problem. In T. Bartz-Beielstein, M.J. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Lecture Notes in Computer Science*, 4771, pages 1-15. Springer, Heidelberg, 2007.

This chapter has been published in Engineering Applications of Artificial Intelligence. The full reference follows:

J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794, 2010.

Chapter III

A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots

A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots

Juan G. Villegas [a,b,c] José Manuel Belenguer [d] Enrique Benavent [d] Antonio Martínez [d] Christian Prins [a] Caroline Prodhon [a]

[a] *Laboratoire d'Optimisation des Systèmes Industriels (LOSI), Institut Charles Delaunay, Université de Technologie de Troyes. BP 2060, 10010 Troyes Cedex, France*

[b] *Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes. A.A. 4976, Bogotá D.C., Colombia*

[c] *Departamento de Ingeniería Industrial, Universidad de Antioquia. A.A. 1226, Medellin, Colombia*

[d] *Departament d'Estadística i Investigació Operativa, Universitat de València. 46100 Burjassot (València), Spain*

Abstract

In the single truck and trailer with satellite depots (STTRPSD) a truck with a detachable trailer based at a main depot has to serve the demand of a set of customers accessible only by truck. Therefore, before serving the customers, it is necessary to detach the trailer in appropriated parking places (called satellite depots or trailer points) where goods are transferred between the truck and the trailer. This problem has applications in milk collection where farms can not be reached using big vehicles. In this work we present an integer programming formulation of the STTRPSD. This formulation is tightened with several families of valid inequalities for which we have developed different (exact and heuristic) separation procedures. Using these elements, a branch-and-cut algorithm for the solution of the STTRPSD has been developed. A computational experiment with public instances from the literature shows that the proposed branch-and-cut algorithm solves consistently problems with up to 50 customers and 10 parking places.

Key words: Branch-and-cut, cutting planes, single truck and trailer routing problem (STTRPSD), truck and trailer routing problem (TTRP), vehicle routing problem (VRP)

Email addresses: jg.villegas64@uniandes.edu.co, juan_guillermo.villegas@utt.fr, jvillega@udea.edu.co (Juan G. Villegas), jose.belenguer@uv.es (José Manuel Belenguer), enrique.benavent@uv.es (Enrique Benavent), antonio.martinez-sykora@uv.es (Antonio Martínez), christian.prins@utt.fr (Christian Prins), caroline.prodhon@utt.fr (Caroline Prodhon).

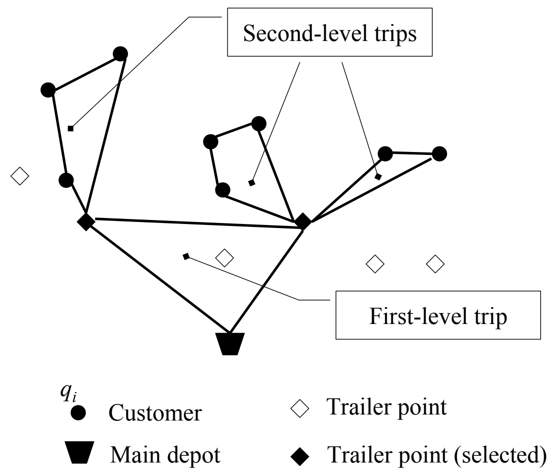


Fig. 1. Example of a STTRPSD solution.

1. Introduction

The single truck and trailer routing problem with satellite depots (STTRPSD) is an extension of the well known vehicle routing problem [11], in which a single vehicle composed of a truck with a detachable trailer serves the demand of a set of customers, reachable only by the truck without the trailer. This accessibility constraint implies the selection of locations to park the trailer before performing the trips to the customers. Therefore, there is a set of parking locations (called trailer points or satellite depots) where it is possible to detach the trailer and to transfer products between the truck and the trailer.

A feasible solution of the STTRPSD (depicted in Figure 1) is composed of: (i) a first-level trip departing from the main depot (performed by the truck with the trailer) visiting a subset of trailer points; and (ii) several second-level trips performed by the truck from any of the trailer points visited in the first level-trip. The goal of the STTRPSD is to minimize the total length of the trips.

Practical applications of the STTRPSD appear mainly in collection and delivery operations in rural areas. For instance, Vanrekkamp [14], Hoff and Løkketangen [9] and Caramia and Guerriero [5] describe milk collection operations where the routes have a STTRPSD-like structure. In that case, the collection is performed by a small tanker with a removable tank trailer of greater capacity. Trailer points are in general parking locations on main roads, while farms are located on narrow roads inaccessible by trailer.

Upper bounds for the STTRPSD have been obtained using a route-first cluster-second procedure embedded within hybrid metaheuristics that combine greedy randomized adaptive search procedures, variable neighborhood descent and evolutionary local search [15,16]. However, to the best of our knowledge, there are no lower bounds or exact approaches for the STTRPSD. Then, the objective of this chapter is to introduce a branch-and-cut algorithm for the solution of the STTRPSD.

The remainder of this chapter is organized as follows. Section 2 presents an integer programming formulation of the STTRPSD. Section 3 introduces several valid inequalities for the STTRPSD that strengthen the initial formulation. Section 4 describes the elements of the proposed branch-and-cut algorithm. Section 5 presents the results of a computational evaluation of the proposed method on a set of instances from the literature. Finally, Section 6 presents some conclusions and outlines future work.

2. Integer programming model

The single truck and trailer routing problem with satellite depots (STTRPSD) can be modeled using an undirected graph $G = (V, E)$, where $V = \{0\} \cup V_d \cup V_c$ is the node set with the main depot at node 0, $V_d = \{1, \ldots, p\}$ is the set of satellite depots (trailer points), and $V_c = \{p+1, \ldots, p+n\}$ is the set of customers with known demands $q_i > 0$ ($\forall i \in V_c$). E is the edge set, each arc (i, j) $(i, j \in V, i < j)$ has a cost c_{ij} which satisfies the triangle inequality. The parameters Q_v and Q_t are the capacities of the truck and the trailer respectively. The total demand of the customers does not exceed $Q_v + Q_t$ to ensure feasibility with one vehicle.

We first introduce some notation. Let $V_1 = \{0\} \cup V_d$ be the set of nodes that can be visited by the truck with the trailer in the first-level trip, and $V_2 = V_d \cup V_c$ be the set of nodes that can be visited by the truck in second-level trips. As usual, $(\forall S \subseteq V)$ $\delta(S)$ denotes the subset of edges with one node in S and other in $V \backslash S$; $\gamma(S)$ is the subset of edges with both nodes in S; and $E(S : S'), \forall S' \subseteq V \backslash S$ denotes the subset of edges with one node in S and other in S'. Finally, the quantity $k(S)$ denotes a lower bound on the number of second-level trips needed to serve the customers in S, $k(S) = \left\lceil \frac{\sum_{i \in S} q_i}{Q_v} \right\rceil$, and $D(S)$ denotes the total demand of the customers in S, $D(S) = \sum_{i \in S} q_i$.

In the mathematical programming formulation, integer variables y_{ij} represent the number of times that the truck with the trailer traverses edge $(i, j) \in \gamma(V_1)$ in the first-level trip. For edges $(i, j) \in \gamma(V_d)$, y_{ij} takes binary values; and for edges $(0, j) \in E(\{0\} : V_d)$ variables y_{0j} may also take the value of 2, representing a direct trip to satellite depot

$j \in V_d$. For a given subset of edges $H \subseteq \gamma(V_1)$, $y(H) = \sum_{(i,j)\in H} y_{ij}$.

Similarly, integer variables x_{ij} represent the number of times edge $(i,j) \in \gamma(V_2)$ is traversed by the truck in second-level trips. For edges $(i,j) \in \gamma(V_c)$, x_{ij} takes binary values; and for edges $(i,j) \in E(V_d : V_c)$ variable x_{ij} may also take the value of 2 when there is a direct trip from a satellite depot $i \in V_d$ to a customer $j \in V_c$. For a given subset of edges $H \subseteq \gamma(V_2)$, $x(H) = \sum_{(i,j)\in H} x_{ij}$.

Finally, in some of the valid inequalities introduced in Section 3, binary variable z_i ($\forall i \in V_d$) states if satellite depot i is selected to park the trailer ($z_i = 1$) or not ($z_i = 0$). Using the notation given above the STTRPSD is formulated as follows:

$$\min \sum_{(i,j)\in\gamma(V_1)} c_{ij} y_{ij} + \sum_{(i,j)\in\gamma(V_2)} c_{ij} x_{ij} \tag{1}$$

Subject to:

$$x(\delta(j)) = 2, \quad \forall j \in V_c \tag{2}$$

$$x(\delta(S)) \geq 2k(S), \quad \forall S \subseteq V_c \tag{3}$$

$$\sum_{i\in I'} x_{ij} + 2x(\gamma(S \cup \{j,l\})) + \sum_{k\in V_d\setminus I'} x_{kl} \leq 2|S| + 3, \quad \forall j,l \in V_c, \forall S \subseteq V_c\setminus\{j,l\}, S \neq \emptyset, \forall I' \subseteq V_d \tag{4}$$

$$\sum_{i\in I'} x_{ij} + 3x_{jl} + \sum_{k\in V_d\setminus I'} x_{kl} \leq 4, \quad \forall j,l \in V_c, \forall I' \subseteq V_d \tag{5}$$

$$\sum_{i\in V_2\setminus S, j\in S_c} x_{ij} + k(S_c) y(\delta(S_d)) \geq 2k(S_c), \quad \forall S_c \subseteq V_c, \forall S_d \subseteq V_d, S = S_c \cup S_d \tag{6}$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in \gamma(V_c) \tag{7}$$

$$x_{ij} \in \{0,1,2\}, \quad \forall (i,j) \in E(V_d : V_c) \tag{8}$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in \gamma(V_d) \tag{9}$$

$$y_{ij} \in \{0,1,2\}, \quad \forall (i,j) \in \delta(\{0\}) \tag{10}$$

The objective function (1) has two terms: the former is the total cost of the first-level trip and the latter is the total cost of the second-level trips. Constraints (2) state that all customers must be visited once and also enforce continuity of second-level trips. Constraints (3) impose the capacity restrictions of second-level trips. Constraints (4), taken from the location-routing problem (LRP) [3], are called path-elimination constraints since they forbid inter-depot second-level trips (i.e. trips that begin at one satellite depot and end at a different one). Constraints (6), called connection constraints, state that each subset of customers $S_c \subseteq V_c$ must be connected to the main depot through one subset of satellite depots $S_d \subseteq V_d$, and that this subset of satellite depots S_d must be connected to

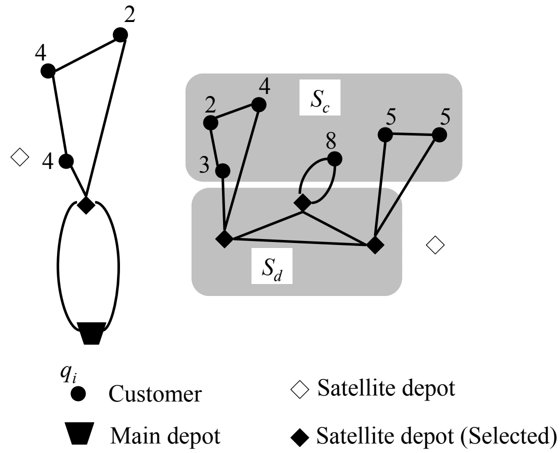


Fig. 2. Example of the violation of a connection constraint ($Q_v = 10$).

the main depot. Constraints (7) to (10) define the variables.

Path-elimination constraints with $S \neq \emptyset$ can be interpreted as follows. When the subtour-elimination constraint $x(\gamma(S \cup \{j, l\})) \leq |S| + 1$ is satisfied with equality all the customers are connected in a single path. Given that this path must be connected to only depots in I' (or $V_d \backslash I'$), if $\sum_{i \in I'} x_{ij} + \sum_{k \in V_d \backslash I'} x_{kl} \geq 1$ constraint (4) will be violated.

To explain path-elimination constraints with $S = \emptyset$, note that when both j and l are served in direct trips $\sum_{i \in I'} x_{ij} = 2$, and $\sum_{k \in V_d \backslash I'} x_{kl} = 2$; and (5) is clearly satisfied because $x_{jl} = 0$. On the contrary, if a single second-level trip serves j and l then $x_{jl} = 1$. In this latter case, $\sum_{i \in I'} x_{ij} + \sum_{k \in V_d \backslash I'} x_{kl} \leq 1$, because the trip must be connected only to depots in I' (or $V_d \backslash I'$), and (5) is also met.

Connection constraints (6) can be explained thanks to their similarity to capacity constraints. If a given $S_c \subseteq V_c$ satisfies the capacity constraint then $x(\delta(S_c)) \geq 2k(S_c)$. But if S_d is composed of the satellite depots from which the trips serving S_c depart, then $\sum_{i \in V_2 \backslash S, j \in S_c} x_{ij} = 0$. Hence, $y(\delta(S_d))$ must be big enough to satisfy (6). This enforces the connection of satellite-depots in S_d to the main depot. Figure 2 illustrates a violated connection constraint.

3. Valid inequalities

In this section we introduce several families of constraints that can be used to strengthen the linear programming relaxation of the STTRPSD formulation given in (1) to (10). Valid inequalities from other vehicle routing problems, like the multi-depot multiple traveling salesman problem (MDMTSP) [4] and the LRP [3], were adapted to the STTRPSD. Whereas, some others were specifically developed for the STTRPSD.

3.1. *Combs with satellite depots*

The multi-depot combs for the MDMTSP introduced by Benavent and Martínez [4] in which each tooth contains at least one depot and the handle contains no depot, can be improved tacking into account the truck capacity. Multi-depot combs for the STTRPSD have the following structure:

$$x(\delta(H)) + \sum_{i=1}^{t} x(\delta(T_i)) \geq 2k(H^1) + 2\sum_{i=1}^{t} k(T_i^1) \tag{11}$$

Where, $H \subseteq V_c$ is the handle and $T_i \subseteq V$ for $i = 1, ..., t$ are the teeth. $T_i^1 = T_i \cap H$ for $i = 1, ..., t$ and $H^1 = H \setminus \bigcup_{i=1}^{t} T_i$. For the handle the following conditions must be met: $H \setminus \bigcup_{i=1}^{t} T_i \neq \emptyset$, and $H \cap T_i \neq \emptyset$ for $i = 1, \ldots, t$. For the teeth the following conditions apply: $T_i \cap V_d \neq \emptyset$ for $i = 1, \ldots, t$; and $T_i \cap T_j = \emptyset, \forall i, j$.

These constraints can be explained as follows. Since $H \cap V_d = \emptyset$, all the customers in the handle must be served in trips that depart from satellite depots outside the handle. The customers in H^1 can be served by trips that depart from any depot in the teeth or by depots in $V_d \setminus \bigcup_{i=1}^{t} T_i$. In the first case, the edges in $\delta(T_i)$ $(i = 1, \ldots, t)$ are crossed at least $2k(H^1)$ times. In the second case, the trips that serve the customers in H^1 use at least $2k(H^1)$ edges in $\delta(H)$. This explains the first term of the RHS (right-hand-side) of (11). For the second term, note that for any tooth i, the customers in T_i^1 can be served by trips that depart from satellite depots in the same tooth or not. In the former case the trips use at least $2k(T_i^1)$ edges in $\delta(H)$. On the other hand, if the trips are not rooted at satellite depots in T_i, the trips must cross the edges in $\delta(T)$ at least $2k(T_i^1)$ times. Since $T_i \cap T_j = \emptyset$ for any pair i, j of teeth, the second term of the RHS of (11) is obtained by summing over all teeth.

3.2. *Co-circuit constraints*

The co-circuit constraints are based on the following property of any STTRPSD solution: in the graph induced by the x variables (or the y variables or both) the number of times that the edges of a cut are used must be even. Co-circuit constraints for the STTRPSD have the following structure:

$$x(\delta(S)\backslash F_c) \geq x(F_c) - |F_c| + 1, \qquad S_c \subseteq V_c, S_d \subseteq V_d, S = S_c \cup S_d, F_c \subseteq E(S_c : V_c\backslash S_c) : |F_c| \text{ is odd} \quad (12)$$

$$y(\delta(S_d)\backslash F_d) \geq y(F_d) - |F_d| + 1, \qquad S_d \subseteq V_d, F_d \subseteq E(S_d : V_d\backslash S_d) : |F_d| \text{ is odd} \quad (13)$$

$$x(\delta(S)\backslash F_c) + y(\delta(S_d)\backslash F_d) \geq x(F_c) + y(F_d) - |F_c| - |F_d| + 1, \qquad S_c \subseteq V_c, S_d \subseteq V_d, S = S_c \cup S_d, \; F_c \subseteq E(S_c : V_c\backslash S_c) \text{ and } F_d \subseteq E(S_d : V_d\backslash S_d) : |F_c| + |F_d| \text{ is odd} \quad (14)$$

The validity of co-circuit constraints with x variables (12) is discussed. The same arguments can be used for co-circuit constraints with y variables (13). Moreover, if (14) is violated then either $|F_c|$ or $|F_d|$ is odd, and then constraint (12) or (13) corresponding to the set of odd cardinality is also violated. Thus, it is only necessary to search for violated constraints of type (12) and (13).

Given a feasible STTRPSD solution $(\overline{x}, \overline{y}, \overline{z})$, note that all the variables associated with F_c are binary. If $\overline{x}(F_c) < |F_c|$, constraints (12) are trivially satisfied because the RHS is negative or zero. When $\overline{x}(F_c) = |F_c|$, since $|F_c|$ is odd, at least one edge from $\delta(S)\backslash F_c$ must be used, and therefore the constraint is also satisfied.

3.3. *Satellite-depot cuts*

This family of constraints are not valid for all STTRPSD solutions, but given that the triangle inequality holds at least one optimal solution has the following property: if two customers i and j have a joint demand that does not exceed Q_v, then both customers are not served in two direct trips departing from the same satellite depot. Generalizing this observation to a subset of customers S such that $D(S) \leq Q_v$ produces the following constraints:

$$x(E(\{i\}:S)) \leq (|S|+1)z_i, i \in V_d, S \subseteq V_c : D(S) \leq Q_v \tag{15}$$

If $z_i = 0$ constraints (15) are trivially satisfied because in that case $x_{ij} = 0\ \forall j \in V_c$. To analyze the validity of constraints (15) when $z_i = 1$, notice that at most one customer in S will be served in a direct trip. Let $S' \subseteq S$ be the subset of customers served in direct trips. If $|S'|$ is even, all the customers can be matched in pairs creating a solution of smaller cost, and then $|S'| = 0$. On the other hand, if $|S'|$ is odd, $|S'|-1$ customers can be matched to produce a solution of smaller cost; and only one customer is served in a direct trip. Therefore, the cardinality of $|S'|$ is at most 1.

If there are no direct trips then $x(E(\{i\}:S)) \leq |S|$, and constraint (15) is satisfied. On the contrary, if there is one direct trip serving customer j then $x(E(\{i\}:S\backslash\{j\}) \leq |S|-1$, and $x_{ij} = 2$. The sum of these two terms is $x(E(\{i\}:S\backslash\{j\}) + x_{ij} \leq |S|-1+2$, or $x(E(\{i\}:S)) \leq |S|+1$, which satisfies constraint (15).

3.4. *Depot-degree constraints*

As defined by Belenguer et al. [3] for the LRP, a solution satisfies the *TI (Triangular inequality) property* if it does not contain two second-level trips whose joint demand is not greater than Q_v based at the same satellite depot. The following family of constraints are valid for STTRPSD solutions satisfying the TI property:

$$x(\gamma(S)) + x(E(\{i\}:S)) - 2z_i \leq |S|-1, i \in V_d, S \subseteq V_c : D(S) \leq Q_v \tag{16}$$

Constraints (16) can be rewritten as:

$$x(\gamma(S \cup i)) - 2z_i \leq |S|-1 \text{ or } x(E(\{i\}:S)) \leq 2z_i - 1 + \frac{1}{2}x(\delta(S))$$

Let (x,y,z) be a feasible solution of the problem. If depot i is closed, $z_i = 0 = x(E(\{i\}:S))$ and constraint (16) is trivially satisfied. Otherwise, $z_i = 1$ and satellite depot i is open. Since the triangle inequality holds and $Q(S) \leq Q_v$, then there is at most one trip rooted at satellite depot i serving only customers in S. If $x(E(\{i\}:S)) = k$ there are two possibilities:

(i) If there is one trip rooted at depot i serving only customers in S, then at least one trip will serve customers in S and customers in $V_c\backslash S$. Then $x(E(S:V_c\backslash S)) \geq k-2$. So, $x(\delta(S)) \geq 2(k-2)+2 = 2k-2$, and the inequality $x(E(\{i\}:S)) \leq 2z_i - 1 + \frac{1}{2}x(\delta(S))$ is satisfied.

(ii) If there is not a trip rooted at satellite depot i serving only customers in S then $x(E(S : V_c\backslash S)) \geq k$. Hence, $x(\delta(S)) \geq 2k$ and the inequality $x(E(\{i\} : S)) \leq 2z_i - 1 + \frac{1}{2}x(\delta(S))$ is also satisfied.

3.5. *Generalized subtour elimination constraints for the first-level trip*

The generalized subtour elimination constraints (GSEC) of the orienteering problem [6] are valid for the first-level trip and have the following form:

$$y(\delta(S)) \geq 2z_i, S \subseteq V_1 : 0 \in S, i \in V_1\backslash S \tag{17}$$

3.6. *CVRP constraints*

There are several polyhedral studies of the capacitated vehicle routing problem (CVRP) that present families of valid inequalities that can be used to strengthen the linear relaxation of different CVRP formulations. Moreover, some of the most successful methods for the solution of the CVRP are based on cutting plane approaches [2,13]. Note that, if all satellite depots are shrunk into a single depot, some of the inequalities of the CVRP can be used to strengthen the linear programming relaxation of the STTRPSD. Specifically, in the branch-and-cut, we employ the CVRP combs described by Lysgaard et al. [13].

4. Branch-and-cut

We have implemented a cutting plane scheme to provide lower bounds for the STTRPSD using the linear relaxation of the formulation given in Section 2 and the valid inequalities presented in Section 3. Moreover, this cutting plane scheme has been embedded within a branch-and-cut algorithm to solve the STTRPSD. A description of the elements of the proposed branch-and-cut algorithm follows.

4.1. *Initial linear relaxation*

The cutting plane method begins with a linear program (LP) that includes the objective function (1), customer-degree constraints (2), variable bounds (7) to (10), and some additional (simple) constraints described below. Capacity, path-elimination and connection constraints were not included because of their exponential number.

The additional constraints added to strengthen the initial lower bound are:

(i) A lower bound on the number of second-level trips that will have any feasible solution (i.e. the capacity constraint for $S = V_c$).

(ii) A set of constraints to ensure that if a trailer point is not visited, it does not have any departing second-level trips:

$$x_{ij} \leq y(\delta(\{i\})), \forall i \in V_d, \forall j \in V_c. \tag{18}$$

(iii) The complementary constraints of the previous ones, i.e. if a trailer point is visited in the first-level trip, it must have departing second-level trips:

$$x(\delta(\{i\})) \geq y(\delta(\{i\})), \forall i \in V_d. \tag{19}$$

(iv) A constraint to state that the first-level trip begins and ends at the main depot:

$$y(\delta(\{0\})) = 2. \tag{20}$$

(v) A set of constraints to express the opening of the trailer points, i.e. if a trailer point is open it must be visited once in the first-level trip:

$$y(\delta(i)) = 2z_i, \forall i \in V_d. \tag{21}$$

These constraints are further disaggregated in the so-called logic constraints of the orienteering problem [6]:

$$y_{ij} \leq z_i, \quad \forall i, j \in V_d \tag{22}$$
$$y_{0i} \leq 2z_i, \quad \forall i \in V_d \tag{23}$$

4.2. *Separation procedures*

At any iteration of the cutting plane, given the current LP optimal solution $(\overline{x}, \overline{y}, \overline{z})$, we search for violated inequalities to be added to the LP formulation. In the separation procedures that follow, the weighted graph $G[\overline{x}]$ induced by edges (i, j) with $\overline{x}_{ij} > 0$ will be called the support graph. Likewise, $G[\overline{x}, \overline{y}]$ is the support graph induced when the variables $\overline{y}_{ij} > 0$ are also considered. In $G[\overline{x}]$ the weight of edges $e \in \gamma(V_2)$ is given by $\overline{x}_e$. Additionally, in $G[\overline{x}, \overline{y}]$ the weight of edges in $\gamma(V_1)$ is given by $\overline{y}_e$.

Several exact and heuristic procedures are employed to separate each family of constraints. Some of them are based on separation procedures for the CVRP [2,13], the MDMTSP [4] and the LRP [3]. On the other hand, for specific STTRPSD constraints it was necessary to design and implement several new separation procedures.

4.2.1. *Capacity constraints*

First, violated capacity constraints (3) are sought with the separation procedures developed by Augerat et al. [2], these procedures comprise: a heuristic that verifies the violation of the capacity constraint for each connected component of the support graph $G[\overline{x}]$; a shrinking heuristic that iteratively shrinks the end points of edges with $\overline{x}_{ij} \geq 1$ and verifies the violation of the capacity constraint of the resulting super-node; and the tabu search procedure by Augerat et al. [1]. Additionally, the heuristic separation procedures developed by Lysgaard et al. [13] are called when all previous procedures fail.

4.2.2. *Path-elimination constraints*

A shrinking heuristic is used to separate path-elimination constraints (4-5). It follows the same principle of the one for capacity constraints; when a super-node is connected to more than one satellite depot the violation of a path-elimination constraint is checked. The exact procedure developed by Belenguer et al. [3] to separate path-elimination constraints is also included in the separation phase of the branch-and-cut algorithm.

Given a pair of customers (j, l), constraint (4) can be decomposed into two terms, the first one: $\sum\limits_{i \in I'} x_{ij} + \sum\limits_{k \in V_d \setminus I'} x_{kl}$ depends only on the selection of I'; and the second one: $2x(\gamma(S \cup \{j, l\}))$ depends on the selection of S. The first term is maximum when $I' = \{i \in V_d : \overline{x}_{ij} \geq \overline{x}_{il}\}$. Now consider the selection of S to maximize the second term. Adding the degree constraints for $S \cup \{j, l\}$ we obtain $2\overline{x}(\gamma(S \cup \{j, l\})) + \overline{x}(\delta(S \cup \{j, l\})) = 2(|S| + 2)$. Therefore, maximizing $2x(\gamma(S \cup \{j, l\}))$ is equivalent to minimizing $\delta(S \cup \{j, l\})$. Consequently, the separation procedure looks for the set S including j and l that minimizes $\delta(S)$.

Using the above arguments, the exact procedure for path-elimination constraints enumerates all the pairs of customers (j, l) for each connected component of the support graph $G[\overline{x}]$. It selects the subset of satellite depots $I' \subseteq V_d$ that maximizes the first term of the LHS (left-hand-side) of (4). To find S, it solves a maximum-flow problem in an auxiliary graph $G'[\overline{x}]$.

To build $G'[\overline{x}]$ it adds, to $G[\overline{x}]$, a source node s connected to j and l with edges of infinite capacity, and a sink node t connected with all satellite depots with edges of infinite capacity. After solving the maximum-flow problem from s to t in $G'[\overline{x}]$, the set S is composed of the nodes reachable from s when the arcs in the minimum cut are eliminated and j and l discarded. Given the maximum values of these two terms, it is easy to check the violation of constraint (4) for customers $j, l \in V_c$, and subsets $I' \subseteq V_d$ and $S \subseteq V_c$.

4.2.3. *Connection constraints*

For connection constraints (6) four different separation procedures are applied. Since the connection constraints are in some sense similar to the capacity constraints, the first separation procedure is a heuristic that reuses the subsets $S_c \subseteq V_c$ generated during the capacity-constraint separation for which the capacity constraint is met. Given S_c, a greedy interchange heuristic looks for a subset $S_d \subseteq V_d$ maximizing the violation of the associated connection constraint. The second separation heuristic verifies the violation of connection constraint for each connected component of the support graph $G[\overline{x}, \overline{y}]$.

The third procedure is an exact algorithm for the special case when $k(S_c) = 1$. In that case the maximum flow from each customer to the main depot is found in the support graph $G[\overline{x}, \overline{y}]$. If the maximum flow is smaller than 2 there is a violated connection constraint. $S = S_c \cup S_d$ is composed of the nodes reachable from the customer after removing the edges in the minimum cut.

Finally, a simple tabu search (with only short-term memory) solves the following optimization problem: $\min \zeta(S) = \sum_{i \in V_2 \setminus S, j \in S_c} \overline{x}_{ij} + k(S_c)\overline{y}(\delta(S_d)) - 2k(S_c)$, for $S = S_d \cup S_c$. During the execution of the tabu search, when a given subset S' with $\zeta(S') < 0$ is found the corresponding connection constraint is violated and thus added to the LP.

4.2.4. *Co-circuit constraints*

The heuristic for the separation of co-circuit constraints (12-13) is based on the approach developed by Ghiani and Laporte to separate these constraints for the undirected rural postman problem [7]. Only the procedure for co-circuit constraints (12) is discussed. The separation of (13) is similar.

Note that co-circuit constraints can be rewritten as:

$$\sum_{e \in \delta(S) \setminus F_c} x_e + \sum_{e \in F_c} (1 - x_e) \geq 1 \tag{24}$$

$$S_c \subseteq V_c, S_d \subseteq V_d, S = S_c \cup S_d, F_c \subseteq E(S_c : V_c \setminus S_c) : |F_c| \text{ is odd}$$

For a given $S \subseteq V_d \cup V_c$ the LHS of (24) is maximum if $F_c = \{e \in E(S_c : V_c \setminus S_c) : \overline{x}_e \geq 0.5\}$. When $|F_c|$ is even, either one edge from F_c is removed, or one edge from $\delta(S_c) \setminus F_c$ is added to F_c. The selected edge will be the one that produces the smaller variation of the LHS of (24). The co-circuit constraint associated with S and F_c will be violated if the LHS is smaller than 1. At each iteration of the cutting plane, all sets $S = \{v\} : v \in V_d \cup V_c$ are checked with the heuristic procedure.

An exact procedure based on the separation of blossom inequalities developed by Letchford et al. [12] is also used for the separation of co-circuit constraints. This procedure build the minimum-cut tree [8] T of a graph G where the nodes are $V_c \cup V_d$ and the edges have weights defined as $w_e = \min\{\overline{x}_e, 1 - \overline{x}_e\}$. Let S^f be the subset of nodes defined by the minimum-weight cut associated with edge f of T. The weight of the cut is equal to the LHS of (24). If the weight of a given each f is smaller than 1 the heuristic procedure described above is applied to S^f.

4.2.5. *MDMTSP combs and CVRP combs*

Violated multi-depot combs are identified with the separation procedures developed by Benavent and Martínez [4] for the MDMTSP, in which a set of promising teeth is firstly found using each satellite depot as a seed and expanding the tooth looking for a cut of value smaller than 2. Once a set of promising teeth has been found the procedure seeks an appropriate handle. Whereas, the heuristic procedures of Lysgaard et al. [13] are used to separate CVRP combs.

4.3. *Satellite-depot cuts and depot-degree constraints*

A greedy heuristic is used for the separation of satellite-depot cuts (15). For each open satellite depot $i \in V_d : z_i > 0$, it adds iteratively to S the customer $j^* = \underset{j \in V_c \setminus S : \overline{x}_{ij} - \overline{z}_i > 0}{\operatorname{argmax}} \{\overline{x}_{ij} - \overline{z}_i\}$ provided the total demand of S does not exceed Q_v. Whereas, the separation procedure for depot-degree constraints (16) reuses the subsets $S \subseteq V_c : D(S) \leq Q_v$ generated during the separation of capacity constraints and evaluates all satellite depots trying to find a violated depot-degree constraint.

4.4. *Generalized subtour elimination constraints*

The separation procedure of generalized subtour elimination constraints (17) follows the approach of Fischetti et al. [6]. This procedure finds the maximum flow from the main depot to each open satellite depot ($i \in V_d : z_i > 0$), in the support graph $G[\overline{y}]$ induced by the the edges $e \in \gamma(V_1)$ with $\overline{y}_e > 0$. If the maximum flow is smaller than $2z_i$ a violated generalized subtour elimination constraints has been found. The subset S is composed of the satellite depots reachable from the main depot when the arcs in the minimum cut are removed.

4.5. *Branch-and-cut overview*

The integer programming formulation described in Section 2 and the valid inequalities of Section 3 are used in the branch-and-cut algorithm in the following way. The branch-and-cut algorithm begins with the initial formulation at the root node. Given the current optimal solution of the LP, the separation procedures search for violated inequalities. After the addition of new violated constraints, the LP is reoptimized using the dual simplex algorithm. The cycle of optimization and identification of violated constraints iterates until no more violated inequalities are found. When this happens the strong branching strategy of CPLEX [10] is used to explore the branch-and-cut tree, and the separation procedures are called again at each node of the tree to identify additional violated constraints. The separation algorithms are called sequentially and when one of them finds violated constraints the search is halted and the LP reoptimized. The search for violated constraints also stops when more than 100 violated constraints of any type are found.

When calling the separation procedures, the following strategy is used: first we call the procedures to identify violated capacity constraints. The sets generated during the search are reused to test the possible violation of depot-degree constraints and connection constraints. Then, if no violated constraints of these three types are found, the separation procedure for multi-depot combs is called. If this procedure fails, violated CVRP combs are sought. Next, the separation procedures for the following constraints are called sequentially: connection constraints, satellite-depot cuts, path-elimination constraints, and generalized subtour elimination constraints. The heuristic to separate co-circuit constraints is called always at the end of any iteration. Finally, the exact separation of co-circuit constraints is called when all previous identification procedures fail.

5. Computational experiments

The branch-and-cut algorithm has been implemented in Visual C++ and calls CPLEX 12.1 for the solution of linear programs. The experiments of this section were run using a computer with an Intel Xeon processor running at 2.67 GHz under Windows 7 Enterprise Edition (64 bits) with 4 GB of RAM. The test bed of 32 instances described in [16] (available at `http://hdl.handle.net/1992/1124`) was used. The size of the instances in the test bed varies from 25 to 200 customers and from 5 to 20 satellite depots. For each problem size there are two levels of truck capacity and two types of instances: one with clustered customers (type *c*) and other with randomly distributed customers (type *rd*). Each instance is named with the convention *STTRP-n-p*-$(0.001 \times Q_v)$-*t*, where *t*, indicates the instance type. For each problem, *Heuristic UB* is the best known solution reported by Villegas et al. [16].

Tables 1 and 2 present the lower bounds obtained in a first computational experiment. Table 1 presents the results for small instances with 25 and 50 customers, while Table 2 presents the results for large instances with 100 and 200 customers. In these tables, column *Cutting plane* reports the results obtained at the root node of the branch-and-cut tree, and column *Partial B&C* presents a lower bound obtained with a partial branch-and-cut in which only y and z variables are declared as integer. The gap with respect to the Heuristic UB ($Gap = \frac{UB-LB}{UB} \times 100$) and the CPU times (in seconds) are included. In all tables, values in bold indicate proven optima.

From Table 1 we can conclude that the partial branch-and-cut is effective for small instances. With this approach 6 out of 8 of the instances with 25 customers were solved optimally, whereas for the STTRPSD with 50 customers only 2 instances were solved to optimality. Likewise, the average gap is reduced from 5.27% for the cutting plane to only 0.34% for partial branch-and-cut, just by making integer the variables of the first-level trip. However, this improvement comes at the price of longer (yet acceptable) running times: while the cutting plane methods needs on average only 7.07 seconds, the partial branch-and-cut needs on average 18.93 seconds.

On the other hand, for the bigger instances of Table 2, it can be seen that only one problem has been solved optimally and that partial branch-and-cut has a large average running time of more than 4 hours. Moreover, for problems STTRP-200-20-1-rd and STTRP-200-20-2-rd, we halted the method after 12 hours of computation. In this set of instances, the average gap increases to 8.75% for the cutting plane and to 3.56% for partial branch-and-cut. Another interesting observation is the greater difficulty of problems with randomly distributed customers. Note that for the same problem size and truck capacity, randomly

Problem		Cutting plane			Partial B&C		
Name	Heuristic UB	LB	Gap(%)	Time (s)	LB	Gap(%)	Time (s)
STTRP-25-5-1-c	405.46	380.67	6.12	1.58	403.58	0.46	3.24
STTRP-25-5-2-c	374.79	354.70	5.36	1.23	**374.79**	0.00	1.48
STTRP-25-5-1-rd	584.03	558.93	4.31	1.19	**584.03**	0.01	2.50
STTRP-25-5-2-rd	508.48	476.30	6.33	0.66	**508.48**	0.00	1.73
STTRP-25-10-1-c	386.45	369.52	4.38	4.52	**386.45**	0.00	6.57
STTRP-25-10-2-c	380.86	352.66	7.40	3.47	**380.86**	0.00	7.25
STTRP-25-10-1-rd	573.96	548.18	4.49	4.76	**573.96**	0.00	10.42
STTRP-25-10-2-rd	506.37	473.14	6.56	1.34	505.31	0.21	8.76
STTRP-50-5-1-c	583.07	557.55	4.38	11.19	**583.07**	0.00	7.35
STTRP-50-5-2-c	516.98	487.67	5.67	11.11	516.52	0.09	13.78
STTRP-50-5-1-rd	870.51	829.13	4.75	4.89	858.96	1.33	21.85
STTRP-50-5-2-rd	766.03	729.67	4.75	9.29	760.67	0.71	22.10
STTRP-50-10-1-c	387.83	374.35	3.48	6.30	385.88	0.50	16.11
STTRP-50-10-2-c	367.01	362.25	1.30	8.19	**367.01**	0.00	14.31
STTRP-50-10-1-rd	811.28	750.60	7.48	19.10	801.60	1.19	100.91
STTRP-50-10-2-rd	731.53	676.58	7.51	24.29	725.17	0.87	64.59
Average			5.27	7.07		0.34	18.93

Table 1
Lower bounds for small instances of the STTRPSD

Problem		Cutting plane			Partial B&C		
Name	Heuristic UB	LB	Gap(%)	Time (s)	LB	Gap(%)	Time (s)
STTRP-100-10-1-c	614.02	575.33	6.30	95.31	606.75	1.18	290.32
STTRP-100-10-2-c	547.44	515.94	5.75	112.97	**547.44**	0.00	190.23
STTRP-100-10-1-rd	1275.76	1131.38	11.32	246.87	1193.01	6.49	699.47
STTRP-100-10-2-rd	1097.28	982.06	10.50	89.39	1063.15	3.11	1271.42
STTRP-100-20-1-c	642.61	616.51	4.06	706.10	640.03	0.40	1597.01
STTRP-100-20-2-c	581.56	555.12	4.55	496.48	579.55	0.35	2033.11
STTRP-100-20-1-rd	1143.10	1008.27	11.80	538.48	1091.78	4.49	6607.39
STTRP-100-20-2-rd	1060.75	938.23	11.55	676.02	1031.71	2.74	42885.16
STTRP-200-10-1-c	822.52	746.09	9.29	1481.72	780.35	5.13	6479.10
STTRP-200-10-2-c	714.33	641.06	10.26	946.10	693.05	2.98	27556.55
STTRP-200-10-1-rd	1761.10	1546.84	12.17	3718.61	1619.49	8.04	10335.88
STTRP-200-10-2-rd	1445.94	1310.74	9.35	1853.74	1399.30	3.23	12233.73
STTRP-200-20-1-c	909.46	838.08	7.85	685.63	889.28	2.22	31290.81
STTRP-200-20-2-c	815.51	783.86	3.88	466.10	810.27	0.64	7460.43
STTRP-200-20-1-rd	1614.18	1426.86	11.60	4166.60	1465.47	9.21	43219.42
STTRP-200-20-2-rd	1413.32	1274.19	9.84	4237.81	1318.63	6.70	43302.70
Average			8.75	1282.37		3.56	14840.79

Table 2
Lower bounds for large instances of the STTRPSD

distributed problems have in general larger gaps and longer running times than those of clustered problems.

In a second computational experiment we tested the full branch-and-cut algorithm (i.e. with all the variables as integer). A running time limit of 30 minutes for the small instances and 3 hours for the large ones has been added. Tables 3 and Table 4 present for each problem, the value of the upper bound found with the multi-start evolutionary local search of [16] and the respective gap with respect to the lower bound found by the branch-and-cut algorithm. For the branch-and-cut, these tables present the lower and upper bounds found, and the gap between them. The number of nodes in the branch-and-cut tree (*Nodes*), the total running time of the algorithm (*Time*), the distribution of this time between the solution of the linear relaxations (*Time LP*) and the separation procedures (*Time Sep.*), as well as the number of cuts found (*# Cuts*). As can be seen in Table 3, all small instances were solved optimally. In particular, the branch-and-cut algorithm solved problems STTRP-25-5-1-c and STTRP-25-10-2-rd and six more 50-customer instances not solved optimally with partial branch-and-cut. It is worth noting that the best solutions found by multi-start evolutionary local search [16] were optimal in all these problems.

Table 4 summarizes the results of the branch-and-cut algorithm for large instances. As can be seen, three more 100-customer instances were solved optimally, all of them of type *c*. The bounds of the partial branch-and-cut were not improved on problems with 100 customers of type *rd* and on problems with 200 customers. This result is explained by the fact that the branch-and-cut tree is explored differently (when the x variables are integer), and because the time limit of 3 hours is shorter than the average time of the partial branch-and-cut. Moreover, for many of these problems the full branch-and-cut has large gaps or did not find feasible solutions. Additional developments are needed to solve these instances.

We performed an additional experiment to evaluate the contribution of each valid inequality to the results of the branch-and-cut algorithm. Seven variants of the cutting plane algorithm were used. The base version only includes the separation procedures for capacity (3), path-elimination (4-5) and connection constraints (6), i.e., the constraints of the initial formulation. The other six variant were obtained by adding, to the base version, the separation procedures of each family of inequalities. Table 5 presents the results of this experiment where each variant of the cutting plane was used to strengthen the lower bound at the root node. For each variant, Table 5 reports the gap with respect to the upper bound ($Gap_{version} = \frac{UB-LB_{version}}{UB} \times 100$), and the gap closed with respect to the lower bound of the base version ($\textit{Closed gap}_{version} = \frac{LB_{version}-LB_{base}}{UB-LB_{base}} \times 100$).

As can be seen in Table 5, with the exception of GSEC, all the constraints reduce the gap of the base version, the largest improvement has been obtained with MDMTSP combs, followed by co-circuit constraints and depot-degree constraints. The result of the GSEC was expected given their structure. In general, y variables have small values at the root node and to have a violation of (17) these variables must be greater. Motivated by the re-

Problem			Branch-and-cut							
Name	Heuristic UB	Gap(%)	LB	UB	Gap (%)	Nodes	Time (s)	Time LP (s)	Time Sep. (s)	# Cuts
STTRP-25-5-1-c	**405.46**	0.00	405.46	**405.46**	0.00	26	2.99	1.36	1.64	324
STTRP-25-5-2-c	**374.79**	0.00	374.79	**374.79**	0.00	5	1.45	0.67	0.78	200
STTRP-25-5-1-rd	**584.03**	0.00	584.03	**584.03**	0.00	6	1.59	0.66	0.92	294
STTRP-25-5-2-rd	**508.48**	0.00	508.48	**508.48**	0.00	75	5.82	2.77	3.04	472
STTRP-25-10-1-c	**386.45**	0.00	386.45	**386.45**	0.00	9	5.93	1.78	4.15	399
STTRP-25-10-2-c	**380.86**	0.00	380.86	**380.86**	0.00	15	6.43	2.12	4.31	461
STTRP-25-10-1-rd	**573.96**	0.00	573.96	**573.96**	0.00	23	9.50	2.80	6.69	616
STTRP-25-10-2-rd	**506.37**	0.00	506.37	**506.37**	0.00	32	6.07	2.88	3.18	468
STTRP-50-5-1-c	**583.07**	0.00	583.07	**583.07**	0.00	9	13.94	3.75	10.19	795
STTRP-50-5-2-c	**516.98**	0.00	516.99	**516.99**	0.00	14	22.05	5.08	16.97	900
STTRP-50-5-1-rd	**870.51**	0.00	870.51	**870.51**	0.00	507	368.90	230.78	138.12	4178
STTRP-50-5-2-rd	**766.03**	0.00	766.03	**766.03**	0.00	89	66.06	22.06	43.99	1609
STTRP-50-10-1-c	**387.83**	0.00	387.83	**387.83**	0.00	141	40.74	17.07	23.67	1162
STTRP-50-10-2-c	**367.01**	0.00	367.01	**367.01**	0.00	8	13.47	2.84	10.62	502
STTRP-50-10-1-rd	**811.28**	0.00	811.28	**811.28**	0.00	661	682.68	439.99	242.69	5417
STTRP-50-10-2-rd	**731.53**	0.00	731.53	**731.53**	0.00	46	87.74	22.30	65.44	2029
Average		0.00			0.00		83.46	47.43	36.03	

Table 3
Results of the branch-and-cut algorithm for small instances of the STTRPSD

Problem			Branch-and-cut							
Name	Heuristic UB	Gap(%)	LB	UB	Gap (%)	Nodes	Time (s)	Time LP (s)	Time Sep. (s)	# Cuts
STTRP-100-10-1-c	**614.02**	0.00	614.02	**614.02**	0.00	5264	7011.09	5222.04	1789.05	7069
STTRP-100-10-2-c	**547.44**	0.00	547.44	**547.44**	0.00	74	359.71	101.92	257.79	2476
STTRP-100-10-1-rd	1275.76	6.74	1189.82	1835.92	35.19	650	10803.94	8465.60	2338.35	21618
STTRP-100-10-2-rd	1097.28	7.45	1015.54	1232.58	17.61	640	10801.04	8067.27	2733.78	18743
STTRP-100-20-1-c	**642.61**	0.00	642.61	**642.61**	0.00	812	2588.79	1059.11	1529.68	5459
STTRP-100-20-2-c	**581.56**	0.00	581.56	**581.56**	0.00	821	5796.84	2922.72	2874.12	7879
STTRP-100-20-1-rd	1143.10	9.82	1030.89	1944.12	46.97	490	10800.74	7408.01	3392.73	15490
STTRP-100-20-2-rd	1060.75	10.64	947.91	1732.57	45.29	560	10801.23	7183.89	3617.34	15624
STTRP-200-10-1-c	822.52	8.21	755.00	a.	-	99	10803.84	5544.12	5259.71	9281
STTRP-200-10-2-c	714.33	10.07	642.40	a.	-	44	10813.28	6878.14	3935.14	5854
STTRP-200-10-1-rd	1761.10	11.52	1558.25	a.	-	54	10808.53	6550.17	4258.36	11734
STTRP-200-10-2-rd	1445.94	8.60	1321.65	a.	-	59	10814.56	7348.22	3466.34	6727
STTRP-200-20-1-c	909.46	7.51	841.16	1053.35	20.14	597	10804.29	6398.23	4406.06	11643
STTRP-200-20-2-c	815.51	2.18	797.75	840.87	5.13	826	10808.75	6259.76	4548.99	8692
STTRP-200-20-1-rd	1614.18	11.16	1434.11	a.	-	35	10800.95	4111.64	6689.31	6707
STTRP-200-20-2-rd	1413.32	9.36	1280.98	a.	-	18	10822.93	3845.26	6977.68	5792
Average		6.45			17.03		9090.03	5460.38	3629.65	

a. No feasible solution has been found by the branch-and-cut algorithm

Table 4
Results of the branch-and-cut algorithm for large instances of the STTRPSD

sults of this experiment, we tried a branch-and-cut without the separation procedures for GSEC. This version solves optimally the same problems of the original version with comparable running times in most of the problems, but on instances STTRP-50-10-1-rd and STTRP-100-20-1-c the running times increased by a factor of 1.72 and 3.22, respectively.

Problem	Base version	MDMTSP combs		CVRP combs		Satellite-depot cuts		GSEC		Co-circuit		Depot-degree	
Name	Gap	Gap	Closed gap	Gap	Closed gap	Gap	Closed gap	Gap	Closed gap	Gap	Closed gap	Gap	Closed gap
STTRP-25-5-1-c	9.85	8.45	14.18	9.85	0.00	9.80	0.55	9.85	0.00	9.08	7.80	9.42	4.37
STTRP-25-5-2-c	9.60	9.14	4.79	9.60	0.00	9.56	0.49	9.60	0.00	9.56	0.43	7.80	18.82
STTRP-25-5-1-r	5.07	4.62	8.92	5.00	1.41	4.84	4.58	5.07	0.00	4.66	8.09	5.04	0.58
STTRP-25-5-2-r	6.99	6.37	8.98	6.76	3.38	6.77	3.18	6.99	0.00	6.58	5.91	6.79	2.93
STTRP-25-10-1-c	5.87	5.13	12.74	5.86	0.29	5.78	1.57	5.87	0.00	5.59	4.87	5.65	3.84
STTRP-25-10-2-c	9.13	8.44	7.55	9.13	0.00	8.93	2.12	9.13	0.00	9.02	1.16	8.67	5.01
STTRP-25-10-1-r	6.11	5.03	17.76	6.11	0.00	6.03	1.34	6.11	0.00	5.55	9.12	5.65	7.47
STTRP-25-10-2-r	7.70	6.62	14.02	7.70	0.00	7.65	0.70	7.70	0.00	6.99	9.24	7.68	0.34
STTRP-50-5-1-c	6.89	6.48	6.00	6.86	0.43	6.88	0.19	6.89	0.00	6.61	4.10	5.68	17.64
STTRP-50-5-2-c	9.66	9.40	2.73	9.66	0.06	9.55	1.20	9.66	0.00	9.28	4.02	7.45	22.96
STTRP-50-5-1-r	5.56	5.40	2.74	5.40	2.85	5.00	9.94	5.56	0.00	5.40	2.84	5.38	3.24
STTRP-50-5-2-r	5.77	5.31	8.07	5.19	10.00	5.60	2.94	5.77	0.00	5.13	11.19	5.82	-0.85
STTRP-50-10-1-c	4.88	4.21	13.72	4.88	0.00	4.77	2.42	4.88	0.00	4.47	8.57	4.26	12.79
STTRP-50-10-2-c	3.07	2.11	31.26	2.96	3.60	3.05	0.67	3.07	0.00	2.23	27.41	2.72	11.25
STTRP-50-10-1-r	8.90	8.25	7.34	8.62	3.17	8.69	2.44	8.90	0.00	8.26	7.25	8.69	2.43
STTRP-50-10-2-r	8.80	7.95	9.60	8.27	5.99	8.70	1.08	8.80	0.00	7.94	9.81	8.72	0.89
STTRP-100-10-1-c	7.83	7.64	2.47	7.72	1.40	7.33	6.47	7.83	0.00	7.60	2.99	6.89	12.00
STTRP-100-10-2-c	7.37	6.57	10.84	7.10	3.64	7.11	3.50	7.37	0.00	6.69	9.19	7.03	4.68
STTRP-100-10-1-r	12.60	12.36	1.95	12.45	1.19	11.60	7.91	12.60	0.00	12.31	2.27	12.06	4.28
STTRP-100-10-2-r	11.80	11.48	2.77	11.40	3.44	11.23	4.87	11.80	0.00	11.23	4.85	11.37	3.71
STTRP-100-20-1-c	5.75	5.15	10.37	5.57	3.01	5.50	4.20	5.75	0.00	4.84	15.70	5.37	6.60
STTRP-100-20-2-c	5.78	5.13	11.22	5.67	1.86	5.57	3.57	5.78	0.00	5.05	12.54	5.52	4.49
STTRP-100-20-1-r	13.22	12.29	7.08	13.07	1.18	12.69	4.03	13.22	0.00	12.16	8.01	13.22	0.06
STTRP-100-20-2-r	12.31	11.85	3.78	12.08	1.86	12.22	0.74	12.31	0.00	11.87	3.56	12.24	0.58
STTRP-200-10-1-c	10.50	10.42	0.73	10.40	0.91	9.66	7.95	10.50	0.00	10.35	1.39	9.83	6.33
STTRP-200-10-2-c	11.31	11.16	1.38	11.05	2.31	10.81	4.48	11.31	0.00	11.03	2.52	11.17	1.23
STTRP-200-10-1-r	14.21	14.15	0.45	14.07	1.00	12.86	9.54	14.21	0.00	14.10	0.78	13.11	7.76
STTRP-200-10-2-r	10.64	10.52	1.10	10.43	1.92	9.75	8.35	10.64	0.00	10.43	1.96	10.37	2.56
STTRP-200-20-1-c	8.89	8.69	2.18	8.85	0.37	8.63	2.95	8.89	0.00	8.59	3.37	8.36	5.98
STTRP-200-20-2-c	4.81	4.47	7.13	4.75	1.27	4.71	2.04	4.81	0.00	4.44	7.76	4.48	6.96
STTRP-200-20-1-r	12.77	12.70	0.54	12.70	0.55	12.04	5.73	12.77	0.00	12.66	0.86	12.58	1.46
STTRP-200-20-2-r	10.72	10.31	3.86	10.56	1.46	10.36	3.34	10.72	0.00	10.37	3.28	10.65	0.66
Average	8.57	8.06	7.44	8.43	1.83	8.24	3.60	8.57	0.00	8.13	6.34	8.11	5.72

Table 5. Contribution of each family of valid inequalities to the lower bound at the root node

6. Conclusions and future work

In this chapter we propose a new formulation and several valid inequalities for the single truck and trailer routing problem with satellite depots. Using them we have implemented a cutting plane method that is embedded within a branch-and-cut algorithm capable of solving instances with up to 50 customers and 10 satellite depots, and clustered problems with up to 100 customers and 20 satellite depots. For these problems, the optimality of the best solutions found by multi-start evolutionary local search [16] was proved.

Moreover, for bigger problems with 100 and 200 customers we derive a good lower bound using a partial branch-and-cut algorithm in which the variables of the first-level trip are declared as integers. In order to solve bigger instances, we are currently working on the adaptation of other valid inequalities from the CVRP, the development of new valid inequalities for the STTRPSD, and the improvement of the general branch-and-cut scheme.

References

[1] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, and D. Naddef. Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research*, 106:546–557, 1998.

[2] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report RR- 949-M, ARTEMIS-IMAG, Université. Joseph Fourier, Grenoble, 1995.

[3] J. M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Wolfler-Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 2010. doi: 10.1016/j.cor.2010.09.019.

[4] E. Benavent and A. Martinez. A branch-and-cut algorithm for the multi depot multiple TSP. In *TRISTAN VII: Seventh Triennial Symposium on Transportation Analysis*, pages 70–73, Tromsø, Norway, June 20-25, 2010.

[5] M. Caramia and F. Guerriero. A milk collection problem with incompatibility constraints. *Interfaces*, 40(2):130–143, 2010.

[6] M. Fischetti, J. J. Salazar-González, and P. Toth. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10(2):133–148, 1998.

[7] G. Ghiani and G. Laporte. A branch-and-cut algorithm for the undirected rural postman problem. *Mathematical Programming*, 87(3):467–481, 2000.

[8] R. Gomory and T. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[9] A. Hoff and A. Løkketangen. A tabu search approach for milk collection in western Norway using trucks and trailers. In *TRISTAN VI: The Sixth Triennial Symposium on Transportation Analysis*, Phuket, Tailand, June 10-15, 2007.

[10] IBM-ILOG. CPLEX 12.1. User's Manual., 2010. ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf.

[11] G. Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics*, 54:811 – 819, 2007.

[12] A. N. Letchford, G. Reinelt, and D. O. Theis. A faster exact separation algorithm for blossom inequalities. In G. L. Nemhauser and D. Bienstock, editors, *IPCO*, volume 3064 of *Lecture Notes in Computer Science*, pages 196–205, 2004.

[13] J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, 2004.

[14] R. Vahrenkamp. Transportation logistic in rural setting. Technical Report 5/1989, Department of Economics, University of Kassel, 1989.

[15] J. G. Villegas, A. L. Medaglia, C. Prins, C. Prodhon, and N. Velasco. GRASP/Evolutionary local search hybrids for a truck and trailer routing problem. In *VIII Metaheuristic International Conference (MIC 2009)*, Hamburg, Germany, July 13-14 2009.

[16] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794, 2010.

PART II: Truck and trailer routing problem

Chapter IV

A GRASP with evolutionary path relinking for the truck and trailer routing problem

A GRASP with evolutionary path relinking for the truck and trailer routing problem

Juan G. Villegas [a,b,c] Christian Prins [a] Caroline Prodhon [a] Andrés L. Medaglia [b]
Nubia Velasco [b]

[a] *Laboratoire d'Optimisation des Systèmes Industriels (LOSI), Institut Charles Delaunay, Université de Technologie de Troyes. BP 2060, 10010 Troyes Cedex, France*

[b] *Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes. A.A. 4976, Bogotá D.C., Colombia*

[c] *Departamento de Ingeniería Industrial, Universidad de Antioquia. A.A. 1226, Medellin, Colombia*

Abstract

In the truck and trailer routing problem (TTRP) a heterogeneous fleet composed of trucks and trailers has to serve a set of customers, some only accessible by truck and others accessible with a truck pulling a trailer. This problem is solved using a route-first, cluster-second procedure embedded within a hybrid metaheuristic based on a greedy randomized adaptive search procedure (GRASP), a variable neighborhood search (VNS) and a path relinking (PR). We test PR as a post-optimization procedure, as an intensification mechanism, and within evolutionary path relinking (EvPR). Numerical experiments show that all the variants of the proposed GRASP with path relinking outperform all previously published methods. Remarkably, GRASP with EvPR obtains average gaps to best-known solutions of less than 1% and provides several new best solutions.

Key words: Truck and trailer routing problem (TTRP), greedy randomized adaptive search procedures (GRASP), variable neighborhood search (VNS), path relinking, vehicle routing problem

1. Introduction

In the truck and trailer routing problem [7] a heterogeneous fleet composed of m_t trucks and m_r trailers ($m_r < m_t$) serves a set of customers $N = \{1, \ldots, n\}$ from a main depot, denoted by 0. Each customer $i \in N$ has a non-negative demand q_i; the capacities of the

Email addresses: `jg.villegas64@uniandes.edu.co`, `juan_guillermo.villegas@utt.fr`, `jvillega@udea.edu.co` (Juan G. Villegas), `christian.prins@utt.fr` (Christian Prins), `caroline.prodhon@utt.fr` (Caroline Prodhon), `amedagli@uniandes.edu.co` (Andrés L. Medaglia), `nvelasco@uniandes.edu.co` (Nubia Velasco).

trucks and the trailers are Q_t and Q_r, respectively; and the distance c_{ij} between any two points $i, j \in N \cup \{0\}$ $(i \neq j)$ is known. Some customers with limited maneuvering space or accessible through narrow roads must be served only by a truck, while other customers can be served either by a truck or by a *complete vehicle* (i.e., a truck pulling a trailer). These incompatibility constraints create a partition of N into two subsets: the subset of *truck customers* N_t accessible only by truck; and the subset of *vehicle customers* N_v accessible either by truck or by a complete vehicle. The objective of the TTRP is to find a set of routes of minimum total distance such that: each customer is visited in a route performed by a compatible vehicle; the total demand of the customers visited in a route does not exceed the capacity of the allocated vehicle; and the number of required trucks and trailers is not greater than m_t and m_r, respectively. Being an extension of the well known vehicle routing problem (VRP), the TTRP is NP-Hard. For updated reviews of the VRP and its extensions the reader is referred to the books by Toth and Vigo [53] and Golden et al. [24], and the introductory tutorial by Laporte [30].

A solution of the TTRP may have three types of routes: *pure truck routes* performed by a truck visiting customers in N_v and N_t; *pure vehicle routes* performed by a complete vehicle serving only customers in N_v; and finally *vehicle routes with subtours*. The latter are composed of a main tour performed by the complete vehicle visiting only customers in N_v, and one or more subtours, in which the trailer is detached at a vehicle customer location, to visit (with the truck) one or more customers in N_t and probably some customers in N_v. The parking place of the trailer is called the root of the subtour.

Figure 1 depicts a solution of the TTRP with $m_t = 4, m_r = 3, Q_t = 6$, and $Q_r = 10$. Solid lines represent segments traversed by a complete vehicle and dashed lines represent segments traversed by a truck. Figure 1 illustrates some of the special features of the TTRP: (i) only the vehicle customers visited in the main tour can be used as roots in vehicle routes with subtours; (ii) the total demand of the customers of a subtour must not exceed Q_t; (iii) the total demand of the customers of all subtours visited on a vehicle route with subtours may exceed Q_t (but not $Q_t + Q_r$), because at the root of a subtour it is possible to transfer goods between the truck and the trailer; (iv) several subtours may have the same root; (v) the first customer of a vehicle route with subtours cannot be a truck customer, because in that case the trailer would have been detached at the main depot due to the accessibility constraint of the first customer, giving rise to a pure truck route.

For the solution of the TTRP we present a hybrid metaheuristic based on a greedy randomized adaptive search procedure (GRASP), a variable neighborhood search (VNS) and a path relinking (PR). The remainder of this chapter is organized as follows. Section 2

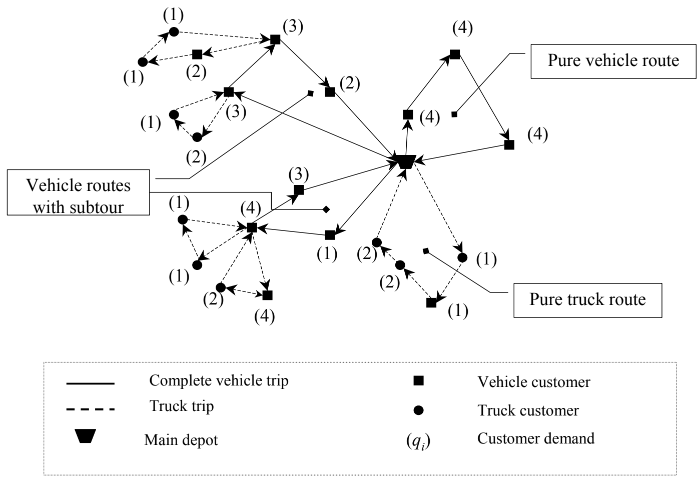


Fig. 1. Feasible solution for the TTRP.

motivates the TTRP with some practical applications and presents the relevant literature on the TTRP and other related problems. Section 3 describes the hybrid metaheuristic and its components. Section 4 presents a computational evaluation of different variants of the hybrid metaheuristic on a set of publicly available test problems and their comparison against other methods from the literature. Finally, Section 5 presents some conclusions. Appendix A summarizes the notation used through the chapter.

2. Literature review

Practical applications of the TTRP appear mainly in collection and delivery operations in rural areas or crowded cities with accessibility constraints. Semet and Taillard [48] used a tabu search to solve a TTRP with time windows, site dependencies and heterogeneous fleet arising in the distribution operations of a chain of grocery stores in Switzerland. Gerdessen [21] described two possible applications of the TTRP. The first one arises in the distribution of dairy products in the Netherlands, where the use of trucks with trailers is common. However, customers located in crowded cities cannot be served by the complete vehicle, thus the trailers must be left in parking lots before reaching these customers. The second application is related with the distribution of compound animal feed in rural regions, customers reachable through narrow roads or bridges must be served by the truck after leaving the trailer parked in a proper place.

Milk collection is another known practical application of the TTRP. Hoff and Løkketangen [28] presented a tabu search algorithm for the solution of a routing problem for milk collection in Norway. They modeled the milk collection using a multi-depot TTRP variant, where N_v is empty and the parking places for the trailers are not associated with customer locations. Likewise, Caramia and Guerriero [6] used the Heterogeneous Milk Collection with Heterogeneous Fleet (HMCHF) problem to model and optimize the milk collection of an Italian dairy company. The HMCHF problem can be seen as a TTRP variant with multi-compartments, route-length constraints, and heterogeneous trucks and trailers.

Chao [7] introduced the TTRP and also proposed a tabu search metaheuristic based on a cluster-first, route-second approach. The clustering phase solves a relaxed generalized assignment problem (RGAP) to allocate customers to routes. The RGAP is solved by rounding the solution of its linear programming relaxation; this rounding may produce infeasible solutions with overcapacity utilization. The second phase uses a cheapest-insertion heuristic to sequence the customers within each route. The insertion heuristic treats pure vehicle routes and pure truck routes as classical traveling salesman problems (TSPs); on the other hand, when constructing vehicle routes with subtours the insertion procedure takes into account the accessibility constraints. In a third step a multiple-neighborhood improvement procedure with a penalized objective function is used to repair infeasible solutions and improve feasible ones. The neighborhoods are reallocations and exchanges of customers between routes, a specialized neighborhood that changes the roots of the subtours, and a 2-opt [18] refinement for every route and subtour. The fourth and final phase is a hybrid tabu search/deterministic annealing method that reuses some of the neighborhoods of the previous phase, and implements a tabu restriction that forbids moves that increase the objective function over a certain threshold. The tabu search has one diversification stage and one intensification stage, executed in sequence. The search is restarted several times from the best solution found so far.

In the same vein, Scheuerer [45] proposed two constructive methods and a tabu search for the TTRP. The first constructive heuristic, called T-Cluster is a cluster-based insertion heuristic that constructs routes sequentially. The insertion of each customer into a route is followed by a steepest descent improvement procedure with three neighborhoods: a root refining approach (for vehicle routes with subtours [7]), 2-opt [18] and Or-opt [33]. The second constructive heuristic, called T-Sweep is an adaptation for the TTRP of the sweep heuristic of Gillet and Miller [22], followed by the same steepest descent procedure. In both methods, it is possible to produce infeasible solutions, because capacity violation of the last route is allowed when there are unrouted customers and no more vehicles available. Starting from the solution obtained with any of the above constructive heuristics, the tabu search method explores the neighborhood generated using reallocations and exchanges of subsets of customers between routes and subtours, and the procedure that changes the roots of vehicle routes with subtours [7]. After the acceptance of a solution, 2-opt and

Or-opt procedures improve each modified tour. The search explores infeasible solutions using a penalized objective function and strategic oscillation following the approach of Cordeau et al. [9]. Neighborhood reduction strategies are used to speed up the evaluation of moves. The search is restarted using the best solution found so far as an intensification mechanism.

Lin et al. [32] developed a very effective simulated annealing (SA) for the TTRP. Their SA uses an indirect representation of the solutions using a permutation of the customers with additional dummy zeros to separate routes and terminate subtours, along with a vector of binary variables of length $|N_v|$, representing the type of vehicle used to serve each vehicle customer (0 for a complete vehicle, and 1 for a truck). A specialized procedure decodes the permutation into a TTRP solution using the information of the binary vector. Since the decoding procedure may fail to find feasible solutions with respect to the availability of trucks and trailers, a route combination approach is used to reduce the number of required trucks and trailers, and within the simulated annealing heuristic a penalty term is added to the objective function to guide the search toward feasible regions. To solve the problem, the authors use a rather standard simulated annealing procedure with three neighborhoods applied to the indirect representation. For the permutation the neighborhoods are: reinsertion of a randomly selected customer or exchange of the position of a random pair of customers; whereas for the binary vector the neighborhood is defined by flipping the type of vehicle serving a randomly selected vehicle customer. To increase the chance of obtaining high quality solutions, half of the time the move performed is the best of several random trials of the selected neighborhood. By relaxing the truck and trailer availability constraints Lin et al. [31] proposed the relaxed TTRP (RTTRP). Reusing their SA, these authors discovered a non-trivial trade-off between the fleet size and the total distance.

Caramia and Guerriero [5] designed a mathematical-programming based heuristic that also employs the cluster-first, route-second approach. Their method solves two subproblems sequentially. The first one, called customer-route assignment problem (CAP) assigns the customers to valid routes seeking to reduce the size of the fleet. Then, given the assignment of customers to routes, the route-definition problem (RDP) minimizes the tour length of each route using a TSP like model without subtour elimination constraints. Since the RDP may produce solutions with disconnected subtours, an edge-insertion heuristic is used to properly connect subtours to the main route. For pure truck routes and pure vehicle routes the heuristic builds a single tour; while for vehicle routes with subtours, the heuristic selects the root of each subtour and constructs the main tour. The authors embedded these two models within an iterative mechanism that adds new constraints to the CAP based on the information of the RDP solution. This restarting mechanism is intended to diversify the search, and includes a tabu search mechanism that forbids (in the CAP) customers route assignments already explored in previous iterations of the algorithm.

In the literature several researchers have studied other vehicle routing problems with trailers that deviate from the TTRP. Semet [47] presented the partial accessibility constrained vehicle routing problem (PACVRP), in which each parking place for the trailer is restricted to have only one departing subtour. He provided an integer programming formulation and developed a cluster-first, route-second approach for the PACVRP. His article mainly discusses the clustering phase modeled with an extended generalized assignment problem, which is solved using Lagrangian relaxation embedded within branch and bound. Gerdessen [21] tackled the vehicle routing problem with trailers (VRPT) using constructive and local search heuristics. The VRPT differs from the TTRP in several simplifying assumptions: in the VRPT there are no accessibility constraints, instead a different service time is incurred if a customer is visited by a complete vehicle or by a truck; all the customers have unit demand, and each trailer is parked exactly once.

Drexl [13] proposed the vehicle routing problem with trailer and transshipments VRPTT. In the VRPTT the customers have time windows; the routing cost is vehicle dependent (i.e., if one arc is traversed by a complete vehicle it has a different cost than if it is traversed just by a truck); parking places (transshipment locations) differ from customer locations; and the assumption of a fixed truck-trailer assignment is dropped, so that a trailer may be pulled by any compatible truck in different routes. The author used a branch-and-price method to solve the VRPTT and showed that only very small instances of this problem can be solved to optimality.

Recently, Villegas et al. [54] have studied the single truck and trailer routing problem with satellite depots (STTRPSD) in which a truck with a detachable trailer based at a main depot has to serve the demand of a set of customers accessible only by truck. Therefore, before serving the customers, the trailer is detached in appropriated parking places (called trailer points of satellite depots) where goods are transferred between the truck and the trailer. To solve the STTRPSD they proposed a multi-start evolutionary local search and a hybrid metaheuristic based on GRASP and variable neighborhood descent (VND). In their computational experiments, on a set of randomly generated instances, multi-start evolutionary local search outperformed GRASP/VND in terms of solution quality and running time.

3. GRASP/VNS with path relinking

GRASP is a memory-less multi-start metaheuristic in which a local search is applied to initial solutions obtained with a greedy randomized heuristic [17]. Even though GRASP has not been widely used for the solution of vehicle routing problems [20], GRASP-based hybrid metaheuristics have achieved competitive results in different routing problems such as the VRP [36], the capacitated location-routing problem [14] and the capacitated arc-routing problem with time windows [39].

Resende and Ribeiro [43] reported that the performance of GRASP can be enhanced by using reactive fine tuning mechanisms, multiple neighborhoods, and path relinking. Along this line, our hybrid metaheuristic includes VNS as the local search component and uses PR in different strategies. A description of the components and the general structure of the hybrid GRASP/VNS with path relinking follows.

3.1. *Greedy randomized construction*

Contrary to most of the solution methods for TTRP-related problems [5–7,28,45,47,48] that use a natural cluster-first, route-second approach, in this chapter we use a route-first, cluster-second (RFCS) procedure for the randomized construction of GRASP. Even though in the 80s Beasley [1] introduced route-first, cluster-second heuristics for the VRP, it was only twenty years later that Prins [35] unveiled its potential as a component of metaheuristics for routing problems. The fundamental idea is to take a giant tour $T = (0, t_1, \ldots, t_i, \ldots, t_n, 0)$ visiting all the customers and break it into VRP feasible routes using a tour splitting procedure. We follow the same spirit for the TTRP.

The randomized route-first, cluster-second heuristic follows three steps. First, a randomized nearest neighbor heuristic with a *restricted candidate list* (RCL) of size κ constructs a giant tour $T = (0, t_1, \ldots, t_i, \ldots, t_n, 0)$, where t_i represents the customer in the i-th position of the tour. Note that T visits all the customers in N, ignoring the capacity of the vehicles and the accessibility constraints of the customers in N_t. Second, we define an auxiliary acyclic graph $H = (X, U, W)$ where the set of nodes X contains a dummy node 0 and n nodes numbered 1 through n, where node i represents customer t_i (i.e., the customer in the i-th position of T); the arc set U contains one arc $(i-1, j)$ if and only if the subsequence $(t_i, \ldots, t_j)$ can be served in a feasible route; and the weight $w_{i-1,j}$ of the arc $(i-1, j)$ is the total distance of the corresponding route. Third, the shortest path between nodes 0 and n in H represents a TTRP solution S, where the cost of the shortest path corresponds to the total distance of S and the arcs in the shortest path represent its routes.

Note that to adapt the route-first, cluster-second (RFCS) approach for the solution of the TTRP it is necessary to take into account its complicating elements, namely, the accessibility constraints and the heterogeneous fixed fleet. We manage the accessibility constraints when building H and take into account the heterogeneous fixed fleet while solving the shortest path on H.

The arc set U in H has three types of arcs, each one representing a type of route. Before adding arc $(i-1, j)$ to U we perform a feasibility test for route $R_{ij} = (0, t_i, \ldots, t_j, 0)$. Let $Q_{ij} = \sum_{u=i}^{j} q_{t_u}$ be the total demand of R_{ij}. If $Q_{ij} < Q_t$ then R_{ij} is feasible (no matter the type of customers assigned to it), and it is a pure truck route. On the other hand, if $Q_t < Q_{ij} \leq Q_t + Q_r$, R_{ij} is feasible if t_i is a vehicle customer, and the type of route represented by arc $(i-1, j)$, depends on the customers assigned to R_{ij}. If all customers belong to N_v then R_{ij} is a pure vehicle route (without subtours). But, if there is at least one customer in N_t, then R_{ij} is a vehicle route with subtours. Finally, if $Q_{ij} > Q_t + Q_r$ the route is infeasible and the arc dropped.

Since the triangle inequality holds, the cost of pure truck routes and pure vehicle routes is easily calculated with $c(R_{ij}) = c_{0t_i} + \sum_{u=i}^{j-1} c_{t_u,t_{u+1}} + c_{t_j 0}$, because its structure corresponds to a single tour. On the contrary, the total distance of vehicle routes with subtours is calculated using an optimization subproblem that selects the parking places for the trailer, and builds the main tour and subtours. This subproblem can be seen as a restricted version of the single truck and trailer routing problem with satellite depots (STTRPSD) [54], in which the satellite depots correspond to the vehicle customers included in R_{ij}.

To solve the restricted STTRPSD associated with the vehicle route with subtours R_{ij} we use a dynamic programming method in which state $[l, m]$ $(i \leq l \leq j, l : t_l \in N_v; l \leq m \leq j)$ represents the use of vehicle customer t_l as the root of a subtour ending at customer t_m. The initial state $[i, i]$ represents the departure of the complete vehicle from the main depot to the first vehicle customer. By definition we include states $[l, l] \forall l : t_l \in N_v$ to represent the movement of the complete vehicle from the root of a subtour to the next vehicle customer performing an empty subtour of null cost.

Let F_{lm} be the cost of state $[l, m]$, and θ_{lkm} be the cost of a subtour $ST = (t_l, t_k, \ldots, t_m, t_l)$ rooted at vehicle customer t_l and visiting customer t_k to t_m. In order to find the parking places for the trailer in route R_{ij} we use a recurrence relation with three cases. The first case represents the initial state ($l = i$ and $m = i$) and corresponds to the trip of the complete vehicle from the depot to the first vehicle customer t_i. Its cost is given by $F_{lm} = c_{0t_i}$. Then, the second case ($l = i$ and $i < m \leq j$) analyses the possibility of performing all the subtours based at vehicle customer t_i. The cost of these states is given by $F_{lm} = \min_k \{F_{ik} + \theta_{i,k+1,m} | k < m : \sum_{u=k+1}^{m} q_{t_u} \leq Q_t\}$.

Finally, the general case $(i < l \leq j, t_l \in N_v; l \leq m \leq j)$ has two terms. In the first term the alternative of having several subtours rooted at vehicle customer t_l is considered, while the second term includes the movement of the complete vehicle from vehicle customer t_h to perform a subtour rooted at vehicle customer t_l. The cost of these states is given by:

$$F_{lm} = \min \left\{ \min_{k} \left\{ F_{lk} + \theta_{l,k+1,m} | k < m : \sum_{u=k+1}^{m} q_{t_u} \leq Q_t \right\}, \min_{h,k} \left\{ F_{hk} + c_{t_h, t_l} + \theta_{l,k+2,m} | h < l : t_h \in N_v; k = l-1 : \sum_{u=k+2}^{m} q_{t_u} \leq Q_t \right\} \right\}$$

In all the cases, the additional conditions over k check the capacity constraints of the truck while performing the subtours; and the conditions over h assure that the main tour only visits vehicle customers. The recursive equation that includes the three cases is presented in Appendix B.

Since states $[l, m]$ do not include the return from the last vehicle customer to the main depot, the cost of the route is finally calculated as $c(R_{ij}) = \min_{i \leq l \leq j : t_l \in N_v} \{F_{lj} + c_{t_l,0}\}$. To solve the STTRPSD we use a similar approach to that used by Villegas et al. [54]. See Appendix B for additional details of the solution procedure.

Once we have generated the auxiliary graph H, we solve a shortest path problem to find the optimal partition of T into a TTRP feasible solution. The limited fleet is taken into account at this stage, thus we solve a resource-constrained shortest path problem (RCSPP) in H, where the resources are the available trucks and trailers. Each arc $(i-1, j)$ in U has three attributes: the distance of the route it represents $w_{i-1,j} = c(R_{ij})$, the consumption of trucks $\alpha_{i-1,j}$, and the consumption of trailers $\beta_{i-1,j}$. The quantities $\alpha_{i-1,j}$ and $\beta_{i-1,j}$ depend on the type of route in the following way: $\alpha_{i-1,j} = 1$, $\beta_{i-1,j} = 0$ if R_{ij} is a pure truck route, while $\alpha_{i-1,j} = \beta_{i-1,j} = 1$ if R_{ij} is a pure vehicle route or a vehicle route with subtours.

In general, shortest path problems with resource constraints can be solved using a generalization of Bellman's algorithm with several labels per node [10]. In our case, let, $\Lambda = (\delta, \tau, \rho, \eta, \lambda)$ be a label associated with any given node $i \in X$ that represents a partial shortest path ending at node i. The label has five attributes: cost δ, truck consumption τ, trailer consumption ρ, father node η, and father label λ. Let $\mathcal{L}_i$ be the set of labels of node i, and let $\Gamma(i)$ be the set of successors of node i, $\Gamma(i) = \{j \in X : (i, j) \in U\}$.

For two labels $\Lambda_1 = (\delta_1, \tau_1, \rho_1, \eta_1, \lambda_1)$, $\Lambda_2 = (\delta_2, \tau_2, \rho_2, \eta_2, \lambda_2)$, we say that Λ_1 dominates in the Pareto sense Λ_2 (denoted $\Lambda_1 \preceq \Lambda_2$) if and only if $\delta_1 \leq \delta_2 \wedge \tau_1 \leq \tau_2 \wedge \rho_1 \leq \rho_2$, and at least one of the inequalities is strict [15]. That is, label Λ_2 is dominated by label Λ_1 because it is possible to reach node j with the same distance and less resource consumption, or with a shorter distance and the same resource consumption.

We use Algorithm 1 to solve the RCSPP. Since by construction H is acyclic, the outmost *for* loop takes the nodes in increasing order, and for a given node i, it scans the set of successors $\Gamma(i)$ (lines 3-23). In the inmost *forall* loop (lines 4-22) all the labels of node i are extended, lines 11-17 perform a domination test, and remove all dominated labels of $\mathcal{L}_j$ if any exists. Finally, line 19 adds non-dominated labels to $\mathcal{L}_j, (j \in \Gamma(i))$. The number of arcs in U is bounded by $O(n^2)$. If we assume that there are no two labels with the same distance, the maximum number of non-dominated labels for each node can be bounded by $O(m_t m_r)$ because m_t and m_r are integers and the consumption is done one unit at a time. Thus, the non-domination test of lines 11-17 is performed in the worst case $O(m_t^2 m_r^2)$ times for each arc. The previous arguments prove that Algorithm 1 runs in $O(n^2 m_t^2 m_r^2)$.

After solving the RCSPP, it is possible to derive the minimum-cost TTRP solution by selecting the label $\Lambda^* = \text{argmin}_{\Lambda \in \mathcal{L}_n} \delta_\Lambda$, (i.e., the cheapest label of node n) and backtracking from it using the information stored in η_{Λ^*}. and λ_{Λ^*}. However, the algorithm for the RCSPP may fail to find a feasible solution and in that case $\mathcal{L}_n = \emptyset$. This occurs when it is not possible to reach node n with at most m_t trucks and m_r trailers. In this case, we relax the fleet-size constraints and solve a classical shortest path problem to find an infeasible solution. The infeasibility of the resulting solution is treated later in the improvement phase and the path relinking procedure.

Figure 2 illustrates the greedy randomized construction for the TTRP. For the sake of clarity we only include in the auxiliary graph the cost of the arcs in the shortest path. The length of each square side in the grid is equal to 1 and the distance between nodes is Euclidean. All customers have unitary demand except customer 1 with $q_1 = 2$. Using the information of the problem (Figure 2(a)) and the sequence of the giant tour (Figure 2(b)), the route-first cluster-second procedure first builds the auxiliary graph (Figure 2(c)). After solving the shortest path problem from node 0 to node 1 in this graph, the route-first, cluster-second procedure builds the TTRP solution of Figure 2(d) using the information of the arcs in the shortest path. Table 1 gives the details of each arc in the auxiliary graph including its tail and head, and the information of the associated route.

Algorithm 1 Labeling algorithm for the resource-constrained shortest path problem

Input: Auxiliary graph H

Output: Shortest path from node 0 to n

Create a label $\Lambda_0 = (0, 0, 0, 0, \emptyset)$; $\mathcal{L}_0 := \mathcal{L}_0 \cup \{\Lambda_0\}$
for $i = 0$ to $n-1$ **do**
    for all $j \in \Gamma(i)$ **do**
        for all $\overline{\Lambda} = (\overline{\delta}, \overline{\tau}, \overline{\rho}, \overline{\eta}, \overline{\lambda}) \in \mathcal{L}_i$ **do**
            $ld := \overline{\delta} + w_{ij}$
            $lt := \overline{\tau} + \alpha_{ij}$
            $lr := \overline{\rho} + \beta_{ij}$
            if $lt \leq m_t$ and $lr \leq m_r$ **then**
                Create a label $\hat{\Lambda} := (ld, lt, lr, i, \overline{\Lambda})$
                $nondom := true$
                for all $\Lambda \in \mathcal{L}_j$ **do**
                    if $\hat{\Lambda} \preceq \Lambda$ **then**
                        $\mathcal{L}_j := \mathcal{L}_j \backslash \{\Lambda\}$
                    else if $\Lambda \preceq \hat{\Lambda}$ **then**
                        $nondom := false$
                    end if
                end for
                if $nondom$ **then**
                    $\mathcal{L}_j := \mathcal{L}_j \cup \{\hat{\Lambda}\}$
                end if
            end if
        end for
    end for
end for

3.2. *Variable neighborhood search for the TTRP*

The improvement phase of the hybrid metaheuristic is a VNS procedure [25]. Taking an initial solution S_0, our VNS performs a classical *variable neighborhood descent* (hereafter VND) step, and then repeats ni main iterations of shaking and improvement alternating between solutions and giant tours. Within VNS we accept infeasible solutions, provided that its infeasibility $\Phi(S)$ does not exceed a given threshold μ. The infeasibility of a given TTRP solution S is calculated using the following expression:

$$\Phi(S) = \max\left\{0, \frac{ut(S)}{m_t} - 1\right\} + \max\left\{0, \frac{ur(S)}{m_r} - 1\right\}$$

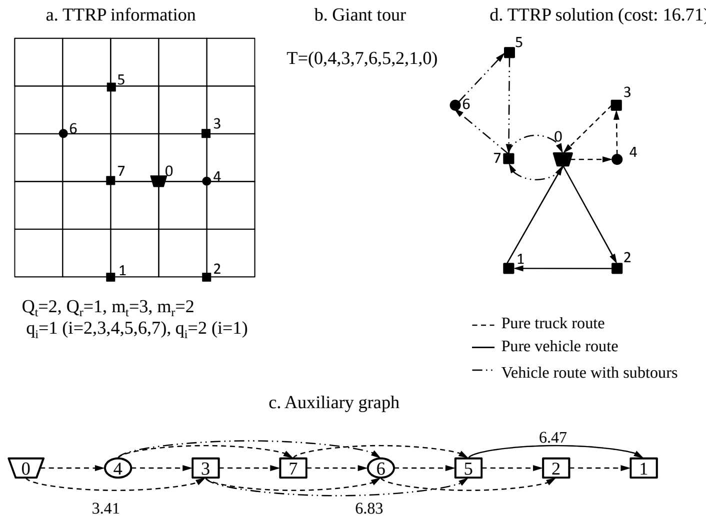


Fig. 2. Example of the route-first cluster-second procedure for the TTRP used in the greedy randomized construction. (a) Problem information; (b) Giant tour; (c) Auxiliary graph; (d) TTRP solution.

Arc		Route				
Tail	Head	Type	Capacity	Load	Cost	Structure
0	4	PTR	2	1	2.00	0-4-0
0	3	PTR	2	2	3.41	(*) 0-4-3-0
4	3	PTR	2	1	2.83	0-3-0
4	7	PTR	2	2	4.65	0-3-7-0
4	6	VRWS	3	3	7.48	0-3-7-0 (Main tour) 7-6-7 (Subtour)
3	7	PTR	2	1	2.00	0-7-0
3	6	PTR	2	2	4.65	0-7-6-0
3	5	VRWS	3	3	6.83	(*) 0-7-0 (Main tour) 7-6-5-7 (Subtour)
7	6	PTR	2	1	4.47	0-6-0
7	5	PTR	2	2	5.89	0-6-5-0
6	5	PTR	2	1	4.47	0-5-0
6	2	PTR	2	2	8.94	0-5-2-0
5	2	PTR	2	1	4.47	0-2-0
5	1	PVR	3	3	6.47	(*) 0-2-1-0
2	1	PTR	2	2	4.47	0-1-0

PTR: Pure truck route, PVR: Pure vehicle route, VRWS: Vehicle route with subtours
(*): Routes of the optimal splitting

Table 1
Information of the arcs of the auxiliary graph for the route-first cluster-second example of Figure 2

where $ut(S)$ and $ur(S)$ represent the number of trucks and trailers required by S. At each call of the VNS, the value of μ is initialized at μ_{max} and decreased after each iteration by $\mu = \mu - \frac{\mu_{max}}{ni}$.

Let $T(S)$ be the giant tour associated with a given TTRP solution $S, T(S)$ is obtained by concatenating all the routes of S in a single string. The shaking procedure works on $T(S)$ by randomly exchanging b pairs of customers with procedure $perturb(T(S), b)$. Then, we derive a new TTRP solution from the perturbed giant tour by using the RFCS approach described above. The value of b is controlled dynamically between 1 and b_{max}, depending on the feasibility and the objective function of the current solution. If the current solution is feasible and updates the best solution visited so far, the value of b is reset to 1 to search in its neighborhood. Whereas, if the current solution is infeasible or the best solution is not improved, b is increased to search in regions far from it. With this mechanism VNS acts as a reparation operator for infeasible solutions and as an improvement procedure for feasible ones.

On the other hand, the procedure $VND(S)$ explores sequentially the following five neighborhoods of a given TTRP solution S using a best-improvement strategy:

– *Modified Or-opt.* For a given chain of customers $(r_i, \ldots, r_{i+l-1})$ of length $l = 1, 2, 3$, check all possible reinsertions of the chain and its reverse $(r_{i+l-1}, \ldots, r_i)$ within the same route or subtour. The difference with classical Or-opt is the simultaneous evaluation of the reversal of the chain.

– *Node Exchange* (in single routes/subtours and between pairs of routes/subtours). Given a pair of customers u and v served by routes (or subtours) R_u and R_v exchange their positions. If $R_u \neq R_v$, in addition to classical capacity constraints, it is necessary to verify the accessibility constraints of u in R_v and v in R_u. Moreover when R_u or R_v is a subtour the capacity of the associated vehicle route is also checked.

– *2-opt* (in single routes/subtours, or pairs of routes (subtours) of the same type). Remove a pair of arcs (u, v) and (w, y) and add two other arcs. For single routes add arcs (u, w) and (v, y); if the arcs belong to different routes we also consider the addition of arcs (u, y) and (w, v) and select the best of the two options. Moreover, if the arcs belong to a pair of subtours we allocate the resulting subtours to the best root among those of the original subtours, provided the capacities of the associated vehicle routes are not exceeded.

– *Node relocation* (in single routes/subtours and between pairs of routes/subtours). Given a customer u served in route/subtour R_u and two consecutive nodes v, w in a route/subtour R', insert u between v and w. If $R_u \neq R'$ we check the conditions for a valid insertion of u in R'.

– *Root refining.* For each subtour we apply the root refining procedure described by Chao

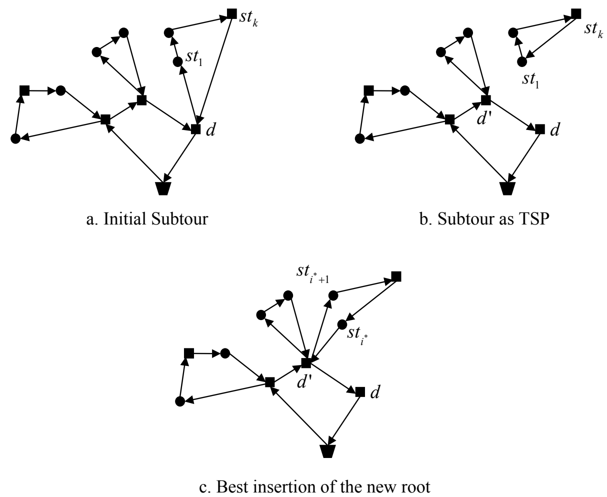


Fig. 3. Example of the root refining neighborhood [7]

[7], where we try to change the root of each subtour and simultaneously modify its routing. Formally, for a subtour $ST = (d, st_1, \ldots, st_k, d)$ the operator removes arcs (d, st_1) and (st_k, d) and adds arc (st_k, st_1) to create a TSP tour. Then the position i^* of the new root d' is found using best insertion and the new subtour becomes $ST' = (d', st_{i^*+1}, \ldots, st_1, st_k, \ldots, st_{i^*}, d')$. To be feasible the new root d' must be served in pure vehicle routes or main tours of vehicle routes with subtours having enough residual capacity to insert the total demand of the subtour. Figure 3 illustrates this neighborhood.

Using the elements described above, Algorithm 2 outlines the VNS component of the proposed metaheuristic.

3.3. *Path relinking*

Path relinking (PR) was introduced in the context of tabu search (TS) as a mechanism that combines intensification and diversification [23] . PR generates new solutions by exploring trajectories connecting elite solutions previously produced during the search. Hybridizing PR with GRASP improves the performance of the latter by tackling the memory-less criticism faced by the basic GRASP scheme [42].

Even though the use of PR in metaheuristics for vehicle routing is rather scarce, it has been applied with relative success. Hybrid metaheuristics combining PR and other methods have been used to solve the classical VRP [27,51], the multi-objective dial-a-ride problem [34], the multi-compartment VRP [16], and the VRP with time windows [26],

Algorithm 2 VNS for the TTRP

Input: Initial solution S_0, **parameters:** μ_{max}, b_{max}, ni
Output: Improved solution S^*

$S_0 := VND(S_0)$
$b := 1$
$\mu := \mu_{max}$
$S^* := \emptyset$
if $\Phi(S_0) = 0$ **then**
    $S^* := S_0$
end if
$S := S_0$
for $i = 1$ to ni **do**
    $T' := perturb(T(S), b)$
    $S' := RFCS(T')$
    $S' := VND(S')$
    if $\Phi(S') \leq \mu$ and $f(S') < f(S)$ **then**
        $S := S'$
    end if
    if $\Phi(S') = 0$ and $f(S') < f^*$ **then**
        $S^* := S'$
        $b := 1$
    else
        $b := \min\{b + 1, b_{max}\}$
    end if
    $\mu = \mu - \frac{\mu_{max}}{ni}$
end for
return S^*

among others. Particularly, GRASP/PR hybrids have been used to solve different routing problems such as the capacitated location-routing problem [38], the team orienteering problem [52], a combined production-distribution problem [3], and the capacitated arc-routing problem with time windows [39], among others.

Resende and Ribeiro [41] give an overview of several ways on how to hybridize PR with GRASP. However, the distance measure, the management of the set of elite solutions and the PR operator are independent from the hybridization mechanism. A brief description of these components for the TTRP follows.

3.3.1. *Distance measure and pool management*

GRASP with PR maintains a pool of elite solutions $\boldsymbol{ES}$. For inclusion in $\boldsymbol{ES}$ a solution S must be better than the worst solution of the pool, but to preserve the diversity of

$\boldsymbol{ES}$, the distance $d(\boldsymbol{ES}, S)$ between S and the pool must be greater than a given threshold Δ, where $d(\boldsymbol{ES}, S) = \min_{S' \in \boldsymbol{ES}}\{d(S, S')\}$. However, the latter condition is overridden when the best solution of $\boldsymbol{ES}$ is updated. With the same diversity objective, S replaces $S_w = min_{S' \in \boldsymbol{ES}: f'(S') > f'(S)} d(S', S)$; i.e., the closer solution in $\boldsymbol{ES}$ that is worst than S. Note that since the pool may contain infeasible solutions, to guide the search toward feasible solutions we use a modified objective function $f' = M_1 \cdot \max\{0, ut(S) - m_t\} + M_2 \cdot \max\{0, ur(S) - m_r\} + f(S)$, where M_1 and M_2 are real numbers such that $M_1 \gg M_2 \gg 0$, and $f(S)$ is the total distance of the routes of S.

Initially $\boldsymbol{ES}$ is filled with $|\boldsymbol{ES}|$ solutions generated with GRASP/VND and $\boldsymbol{ES}$ is always kept ordered according to $f'(S)$. Note that, values for M_1 and M_2 are not explicitly needed because the pool is lexicographically sorted using an order consistent with the one imposed by $f'(S)$. This lexicographic order gives priority to feasible solutions, among feasible solutions to those with smaller distances, and among infeasible solutions to those with smaller infeasibility with respect to the use of trucks, and then to those with smaller infeasibility with respect to the use of trailers.

Different metrics can be used to define the distance between two solutions of vehicle routing problems. For instance, Ho and Gendreau [27] count the number of differing edges; Sörensen and Sevaux [50] measure the distance between trips using the edit distance and then solve a linear assignment problem to match the trips of the solutions. They use the cost of the assignment as the distance between the solutions. Finally, in route-first cluster-second based metaheuristics it is possible to measure the distance between solutions using their corresponding giant tours [37]. In that case, different metrics for distance on permutations could be used [46,49]. Among them, we decided to use the distance for R-permutations [4], also known as the adjacency or broken-pairs distance.

Given two solutions S and S' and their corresponding giant tours $T(S)$ and $T(S')$, the broken-pairs distance counts the number of consecutive pairs that differ from one giant tour to the other, that is $d(T(S), T(S'))$ is defined as the number of times t_{i+1} does not immediately follow t_i in $T(S')$, for $i = 0, \ldots, |N|$. For example, if we have $T(S) = (0, 1, 2, 3, 4, 5, 0)$ and $T(S') = (0, 5, 3, 4, 1, 2, 0)$, the distance $d(T(S), T(S'))$ is 4, because pairs $(0, 1)$, $(2, 3)$, $(4, 5)$ and $(5, 0)$, of $T(S)$ are not in $T(S')$.

3.3.2. *Path relinking operator*

To transform the initial solution S_0 into the guiding solution S_f, the PR operator repairs from left to right the broken pairs of $T(S_0)$, creating a path of giant tours with non-increasing distance to $T(S_f)$. A broken pair is repaired by shifting to the left (in $T(S_0)$) blocks of consecutive customers in such a way that at least one broken pair is repaired without creating new broken pairs. Figure 4 illustrates the PR operator. To increase the chance of finding high quality solutions, the PR operator uses the back-and-forward strategy [41], exploring the forward path from S_0 to S_f, and also the backward path from S_f to S_0. All the giant tours in both paths are split using the route-first cluster-second approach described above to generate a set $\boldsymbol{P}$ of TTRP solutions.

Forward path

	Giant tour	Distance to $T(S_f)$
$T(S_0)$	0 7 6 5 3 4 1 2 0	6
T_1	0 1 2 7 6 5 3 4 0	5
T_2	0 1 2 3 4 7 6 5 0	4
T_3	0 1 2 3 4 5 7 6 0	3
T_4	0 1 2 3 4 5 6 7 0	0
$T(S_f)$	0 1 2 3 4 5 6 7 0	

Backward path

	Giant tour	Distance to $T(S_f)$
$T(S_f)$	0 1 2 3 4 5 6 7 0	6
T_1	0 7 1 2 3 4 5 6 0	5
T_2	0 7 6 1 2 3 4 5 0	4
T_3	0 7 6 5 1 2 3 4 0	3
T_4	0 7 6 5 3 4 1 2 0	0
$T(S_0)$	0 7 6 5 3 4 1 2 0	

Shifting blocks

Fig. 4. Example of the path relinking operator

3.3.3. *Path relinking strategies and overview of the method*

Originally, Laguna and Martí [29] proposed PR as intensification mechanism after each GRASP iteration. Our first GRASP/VNS with PR described in Algorithm 3 follows this approach. In this hybrid method the PR operator explores the paths between a local optimum obtained by GRASP/VNS and a solution randomly chosen from $\boldsymbol{ES}$. The difference with the classical approach is that we apply VND to all feasible solutions produced by the PR operator and test them for insertion in $\boldsymbol{ES}$.

Another possibility is to use the PR operator as post-optimization procedure, after the ns iterations of GRASP/VNS. In this case, each local optimum produced by GRASP/VNS is just checked for inclusion in $\boldsymbol{ES}$. After the main GRASP/VNS loop, the procedure $PathRelinking(\boldsymbol{ES})$, applies PR to all pairs of elite solutions in $\boldsymbol{ES}$ not yet relinked. The resulting feasible solutions ($\boldsymbol{RS}$) are further improved with VND and $\boldsymbol{ES}$ updated. The post-optimization procedure is iterated as long as there are new solutions in $\boldsymbol{ES}$. Algorithm 4 summarizes this variant of GRASP/VNS with PR.

More recently, Resende and Werneck [44] and Resende et al. [40] introduced evolutionary path relinking (EvPR), a variant in which the $PathRelinking(\boldsymbol{ES})$, procedure is used periodically during the search. Consequently, our GRASP/VNS with EvPR for the TTRP (outlined in Algorithm 5) keeps the intensification mechanism, and evolves the elite set

Algorithm 3 GRASP/VNS with PR as intensification mechanism

Input: TTRP, **parameters:** κ, ns, μ_{max}, b_{max}, ni, $|\boldsymbol{ES}|$, Δ
Output: TTRP solution S^*

for $i = 1$ to $|\boldsymbol{ES}|$ **do**
  $T := RandomizedNearestNeighbor(N, \kappa)$
  $S := RFCS(T)$
  $S := VND(S)$
  Insert S in $\boldsymbol{ES}$
end for
for $i = 1$ to ns **do**
  $T := RandomizedNearestNeighbor(N, \kappa)$
  $S := RFCS(T)$
  $S := VNS(S, \mu_{max}, b_{max}, ni)$
  Select at random $S' \in \boldsymbol{ES}$
  $\boldsymbol{P} := PathRelinkingOperator(S, S')$
  for all $\overline{S} \in \boldsymbol{P} : \Phi(\overline{S}) = 0$ **do**
    $\overline{S} := VND(\overline{S})$
    if $d(\boldsymbol{ES}, \overline{S}) \geq \Delta$ **then**
      Try to insert $\overline{S}$ in $\boldsymbol{ES}$
    end if
  end for
end for
$S^* := \text{argmin}_{S \in \boldsymbol{ES}} f(S)$
return S^*

every γ iterations.

4. Computational experiments

We implemented the three variants of the proposed metaheuristic (GRASP/VNS with PR as post-optimization, GRASP/VNS with PR as intensification and GRASP/VNS with EvPR) using Java and compiled them using Eclipse JDT 3.5.1. We ran the experiments of this section on a computer with an Intel Xeon running at 2.67 GHz under Windows 7 Enterprise Edition (64 bits) with 4 GB of RAM. Table 2 summarizes the characteristics of each problem in the 21-instance test bed described by Chao [7], where the size of the problems range from $n = 50$ to $n = 199$ and for each problem size there are three values for the fraction of truck customers (25%, 50% and 75%).

All the variants of GRASP/VNS with PR share the size of the RCL (κ) and number of iterations (ns) of GRASP; the maximum infeasibility threshold (μ_{max}), maximum number of pairs (b_{max}) and number of iterations (ni) of VNS; and the distance threshold (Δ) and size of the elite set ($|\mathbf{ES}|$) of PR. Additionally, EvPR is applied every γ iterations. We also included in the computational experiment a GRASP/VNS (without PR) as a base case (benchmark) to analyze the contributions of PR. After some preliminary experimentation

Algorithm 4 GRASP/VNS with PR as post-optimization mechanism

Input: TTRP, **parameters:** κ, ns, μ_{max}, b_{max}, ni, $|\boldsymbol{ES}|$, Δ
Output: TTRP solution S^*

for $i = 1$ to ns **do**
  $T := RandomizedNearestNeighbor(N, \kappa)$
  $S := RFCS(T)$
  $S := VNS(S, \mu_{max}, b_{max}, ni)$
  if $d(\boldsymbol{ES}, S) \geq \Delta$ **then**
    Try to insert S in $\boldsymbol{ES}$
  end if
end for
$new := true$
repeat
  $\mathbf{ES}_0 := \boldsymbol{ES}$
  $\boldsymbol{RS} := PathRelinking(\boldsymbol{ES})$
  for all $\overline{S} \in \boldsymbol{RS}$ **do**
    if $\Phi(\overline{S}) = 0$ **then**
      $\overline{S} := VND(\overline{S})$
    else
      $\boldsymbol{RS} := \boldsymbol{RS} \backslash \{\overline{S}\}$
    end if
  end for
  $\mathbf{ES} := Update(\mathbf{ES}, \mathbf{RS})$
  if $\mathbf{ES} \backslash \mathbf{ES}_0 = \emptyset$ **then**
    $new := false$
  end if
until $new = false$
$S^* := \text{argmin}_{S \in \boldsymbol{ES}} f(S)$
return S^*

we set the parameters of the different variants of the hybrid metaheuristic to the values summarized in Table 3.

Table 4 presents the best and average results over 10 runs of the GRASP/VNS with PR variants and the GRASP/VNS benchmark. The column labeled *Time* reports the average running time in minutes for each method. We also include the best-known solutions (*BKS*) for each instance, taken from Lin et al. [32] and Scheuerer [45] and updated with some new best-known solutions found by the proposed GRASP/VNS with PR. The last rows of the table summarize the average gap above best-known solutions, the number of times each method found the best know solution (*NBKS*), and the average running time. Values in bold in the table indicate that the BKS was found by a given method.

All the variants of GRASP/VNS with PR largely outperform the base case GRASP/VNS (without PR), highlighting the contribution of PR to the quality of the solutions. Remarkably, the use of PR as a post-optimization mechanism offers a good trade-off between running time and solution quality. The post-optimization with PR approximately halved

Algorithm 5 GRASP/VNS with evolutionary path relinking

Input: TTRP, **parameters:** κ, ns, μ_{max}, b_{max}, ni, $|\boldsymbol{ES}|$, Δ
Output: TTRP solution S^*
for $i = 1$ to $|\boldsymbol{ES}|$ **do**
    $T := RandomizedNearestNeighbor(N, \kappa)$
    $S := RFCS(T)$
    $S := VND(S)$
    Insert S in $\boldsymbol{ES}$
end for
for $i = 1$ to ns **do**
    $T := RandomizedNearestNeighbor(N, \kappa)$
    $S := RFCS(T)$
    $S := VNS(S, \mu_{max}, b_{max}, ni)$
    Select at random $S' \in \boldsymbol{ES}$
    $\boldsymbol{P} := PathRelinkingOperator(S, S')$
    for all $\overline{S} \in \boldsymbol{P} : \Phi(\overline{S}) = 0$ **do**
        $\overline{S} := VND(\overline{S})$
        if $d(\boldsymbol{ES}, \overline{S}) \geq \Delta$ **then**
            Try to insert $\overline{S}$ in $\boldsymbol{ES}$
        end if
    end for
    if $i \bmod \gamma = 0$ **then**
        $new := true$
        repeat
            $\mathbf{ES}_0 := \boldsymbol{ES}$
            $\boldsymbol{RS} := PathRelinking(\boldsymbol{ES})$
            for all $\overline{S} \in \boldsymbol{RS}$ **do**
                if $\Phi(\overline{S}) = 0$ **then**
                    $\overline{S} := VND(\overline{S})$
                else
                    $\boldsymbol{RS} := \boldsymbol{RS} \backslash \{\overline{S}\}$
                end if
            end for
            $\mathbf{ES} := Update(\mathbf{ES}, \mathbf{RS})$
            if $\mathbf{ES} \backslash \mathbf{ES}_0 = \emptyset$ **then**
                $new := false$
            end if
        until $new = false$
    end if
end for
$S^* := \text{argmin}_{S \in \boldsymbol{ES}} f(S)$
return S^*

the average gap to BKS of GRASP/VNS with an increase of only 30% in the running time. Moreover, this variant was able to improve the BKS of problem 15[1]. Above all, GRASP/VNS with EvPR stands as the best performing method, having an average gap to BKS as small as 0.84% obtaining 12 out of 21 BKS, and improving the BKS of problems 11, 12 and 14[1]. However, this outstanding performance is achieved at the price of almost

[1] Detailed solutions available at http://hdl.handle.net/1992/1127

Problem number	Customers			Trucks		Trailers		Demand-Capacity ratio
	n	$\|N_v\|$	$\|N_t\|$	m_t	Q_t	m_r	Q_r	
1	50	38	12					
2	50	25	25	5	100	3	100	0.971
3	50	13	37					
4	75	57	18					
5	75	38	37	9	100	5	100	0.974
6	75	19	56					
7	100	75	25					
8	100	50	50	8	150	4	100	0.911
9	100	25	75					
10	150	113	37					
11	150	75	75	12	150	6	100	0.931
12	150	38	112					
13	199	150	49					
14	199	100	99	17	150	9	100	0.923
15	199	50	149					
16	120	90	30					
17	120	60	60	7	150	4	100	0.948
18	120	30	90					
19	100	75	25					
20	100	50	50	10	150	5	100	0.903
21	100	25	75					

Table 2
Test problems for the TTRP

Method	GRASP		VNS			Path relinking		
	κ	ns	μ_{max}	b_{max}	ni	$\|\mathbf{ES}\|$	Δ	γ
GRASP/VNS	2	60	0.25	6	200	-	-	-
GRASP/VNS with PR (Intensification)	2	60	0.25	6	200	5	$\max\{10, m_t + m_r\}$	-
GRASP/VNS with PR (Post-optimization)	2	60	0.25	6	200	5	$\max\{10, m_t + m_r\}$	-
GRASP/VNS with EvPR	2	60	0.25	6	200	5	$\max\{10, m_t + m_r\}$	20

Table 3
Parameters of the proposed GRASP/VNS with PR variants

doubling the running time of the benchmark GRASP/VNS.

Table 5 presents the comparison of the proposed hybrid metaheuristic against other methods from the literature. In this table we only compare against the best variant, namely GRASP/VNS with EvPR. Table 4 includes the results of the tabu search of Chao [7] and Scheuerer [45], the simulated annealing of Lin et al. [32], and the mathematical-programming based heuristic of Caramia and Guerriero [5]. Depending of the availability of results we report the best and average results over 10 runs of each metaheuristic. For the heuristic of Caramia and Guerriero, we only report the results of a single run of their deterministic method. Column *BKS* presents the best-known solution for each problem reported for the first time in the paper cited in column *Ref*, column *Gap* reports the gap with respect to BKS (in %) for each instance and each method. The last rows of Table 5

Problem			GRASP/VNS			GRASP/VNS +PR (Intensification)			GRASP/VNS +PR (Post-optimization)			GRASP/VNS + EvPR		
Number	n	BKS	Best	Avg.	Time	Best	Avg.	Time	Best	Avg.	Time	Best	Avg.	Time
1	50	564.68	**564.68**	568.31	0.91	**564.68**	566.22	1.07	**564.68**	566.38	0.98	**564.68**	565.99	1.17
2	50	611.53	614.27	617.53	1.00	**611.53**	614.46	1.18	**611.53**	614.81	1.08	**611.53**	614.23	1.29
3	50	618.04	**618.04**	619.07	0.86	**618.04**	618.04	0.98	**618.04**	618.24	0.91	**618.04**	618.04	1.05
4	75	798.53	802.41	815.16	1.86	802.41	805.72	2.37	799.34	805.62	2.08	**798.53**	803.51	2.69
5	75	839.62	841.81	857.79	1.99	**839.62**	844.99	2.59	840.74	848.18	2.19	**839.62**	841.63	2.82
6	75	930.64	989.71	1040.19	1.95	952.43	967.77	2.57	946.66	970.18	2.36	940.59	961.47	2.89
7	100	830.48	830.62	832.27	4.11	**830.48**	830.55	5.55	**830.48**	830.71	4.66	**830.48**	830.48	6.05
8	100	872.56	881.53	885.01	4.35	874.95	878.48	5.99	874.73	879.02	5.12	**872.56**	876.21	6.96
9	100	912.02	916.63	930.55	5.67	915.29	920.22	7.66	915.46	921.39	6.25	914.23	918.45	8.38
10	150	1039.07	1050.76	1062.03	9.95	1047.25	1051.40	16.01	1047.59	1054.40	13.10	1046.71	1050.11	18.84
11	150	1093.37	1114.64	1122.43	11.02	1095.94	1108.45	18.30	1097.75	1105.37	14.63	**1093.37**	1100.95	21.20
12	150	1152.32	1159.88	1174.89	14.00	1155.09	1163.67	23.21	1153.04	1159.11	18.44	**1152.32**	1158.88	25.78
13	199	1287.18	1319.38	1332.55	18.71	1304.77	1314.52	36.33	1301.22	1310.78	26.15	1298.89	1305.83	43.94
14	199	1339.36	1380.86	1395.50	20.07	1357.05	1367.50	39.96	1351.23	1362.02	30.02	**1339.36**	1354.04	45.57
15	199	1420.72	1454.10	1462.23	25.14	1430.38	1443.45	52.53	**1420.72**	1436.29	37.12	1423.91	1437.52	59.83
16	120	1002.49	1003.99	1005.88	8.13	**1002.49**	1003.51	12.12	**1002.49**	1003.82	9.99	**1002.49**	1003.07	14.73
17	120	1026.20	1045.08	1050.86	8.09	1042.53	1042.99	11.86	1042.53	1044.76	9.41	1042.46	1042.61	13.17
18	120	1098.15	1121.07	1128.51	7.73	1114.33	1121.00	11.16	1113.18	1120.02	9.01	1113.07	1118.63	12.69
19	100	813.30	817.11	820.94	3.82	814.73	820.45	4.74	813.72	820.35	4.18	813.50	819.81	5.21
20	100	848.93	860.12	861.34	4.21	860.12	860.12	5.28	860.12	860.12	4.49	860.12	860.12	5.62
21	100	909.06	912.35	913.62	4.59	**909.06**	909.60	5.83	**909.06**	910.33	5.06	**909.06**	909.06	6.31
Avg. gap above BKS			1.29%	2.26%		0.60%	1.11%		0.48%	1.09%		0.36%	0.84%	
NBKS			2			7			7			12		
Avg. Time (min)					7.53			12.73			9.87			14.58

Table 4
Results of the proposed metaheuristics in the test problems of [7]

summarize the average cost over the 21 test problems, the average gap above best-known solutions (*BKS*), and the number of times each method found the best-known solution (*NBKS*).

As can be seen in Table 5, GRASP/VNS with EvPR outperforms all the methods from the literature, achieving a small average gap to BKS of 0.84% and obtaining 12 out of 21 BKS with a single set of parameters. Our method improved the BKS for four large problems. GRASP/VNS with EvPR almost halved the average gap to BKS of the simulated annealing heuristic of Lin et al. [32], the previous best method with an average gap to BKS of 1.51%, and the second-best methods by Scheuerer [45] and Caramia and Guerriero [5], which obtained the same average gap to BKS of 1.74%. Finally, the tabu search of Chao [7] with an average gap to BKS of 7.55% is clearly outperformed by GRASP/VNS with EvPR.

It is important to note that the worst performance of EvPR (as for the other variants of the hybrid metaheuristic) is obtained on problem 6. By analyzing some statistics during the search, we observed that due to the very tight demand to capacity ratio (0.974) it is very difficult to find feasible solutions with the proposed RFCS approach. In contrast, the method by Caramia and Guerriero [5] is better adapted to solve this problem since it has a packing step that produces a feasible solution if any exists.

Some authors used the average cost over the 21 instances as a measure to compare the performance of different metaheuristics for the TTRP [5,32]. Nonetheless, this is not a good measure because it favors methods with good results in the bigger instances. For instance, the cost of the BKS of problem 15 is 2.5 times the cost of the BKS of problem 1. Then an improvement of 1% of the BKS of problem 15 will have 2.5 times more impact in this measure than the same improvement on problem 1. Hence, to have a better comparison of the algorithms, we followed Garcia et al. [19] and used the Friedman test to analyze the average results of the metaheuristics for which there are several replications.

The null hypothesis of the Friedman test is that each ranking of the algorithms within each problem is equally likely, so there is no difference between them. As can be seen in Table 6, GRASP/VNS with EvPR consistently ranks in the first two positions. The Friedman test was performed according to the procedure described by Conover [8] and the analysis led to the rejection of the null hypothesis with a level of significance $\alpha = 1\%$. Moreover, the paired comparisons unveiled that GRASP/VNS with EvPR is better than each one of the other methods with the same level of significance.

Since the method of Caramia and Guerriero [5] is deterministic, only one run is enough to characterize its performance. To have a fair comparison against it, in Table 6 we compared their results against those of the best and worst runs (i.e., the seeds that produce the smallest and biggest average deviations above BKS, respectively). In the last row of Table 6 it is possible to see that GRASP/VNS with EvPR is consistently better than Caramia and Guerriero's method in 13 out of 21 problems, regardless of the seed. Even though, the best seed has a much smaller gap to BKS of 0.69% compared to 1.74% of their method, the worst seed is still better than their method with a gap to BKS of 1.05%. This experiment also highlights the robustness of GRASP/VNS with EvPR, that is, the performance of a single run is very close to the average performance.

To compare the running times of GRASP/VNS with EvPR against those reported in the literature we chose the tabu search of Scheuerer [45] and the simulated annealing of Lin et al. [32]. The tabu search of Chao [7] was discarded because it has a large gap to BKS and, unfortunately, Caramia and Guerriero [5] did not report running times.

To have a fair comparison of the running times we scaled the time spent by GRASP/VNS with EvPR to the reference computer used by Scheuerer [45] and Lin et al. [32]. Both of

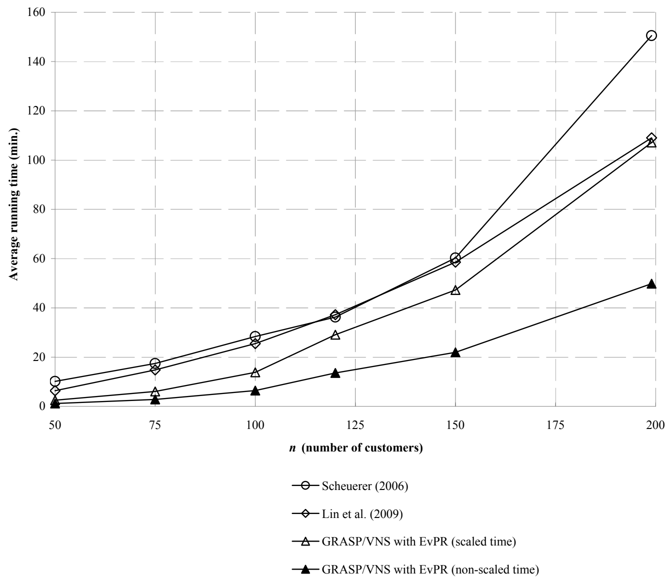


Fig. 5. Comparison of average running time of GRASP/VNS with EvPR and other published methods

them used an Intel Pentium IV PC running at 1.5GHz. Scheuerer [45] reported a speed factor of around 326Mflop/s (millions of floating-point operations per second) for this computer using the Linpack benchmark [11]. Using the Java version of the Linpack benchmark [12], we obtained a speed factor of approximately 702 Mflop/s for our Intel Xeon running at 2.67 GHz. Using these values we derived a scaling factor of 2.15 for our running times.

Table 8 shows that GRASP/VNS with EvPR has in general shorter running times. The column labeled *Speed-up* reports the ratio of the running times of the published algorithms over the scaled time of GRASP/VNS with EvPR. Following the arguments outlined by Bixby [2], we report in the last row the geometric mean of these ratios as a conservative estimate of the speed-up factor achieved by GRASP/VNS with EvPR with respect to the tabu search of Scheuerer [45] and the simulated annealing of Lin et al. [32]. The comparison of Table 8 is further illustrated through Figure 5. However this running time comparison must be taken with care since the operating system, computer configuration, and programming language varies for each method.

Problem				Chao [7] (Tabu Search)		Scheuerer [45] (Tabu Search)				Lin et al. [32] (Simulated Annealing)				Caramia and Guerriero [5] (Math. Prog. Heuristic)		GRASP/VNS + EvPR			
Number	n	BKS	Ref.	Avg. Cost	Gap	Best cost	Gap	Avg. Cost	Gap	Best cost	Gap	Avg. Cost	Gap	Cost	Gap	Best cost	Gap	Avg. Cost	Gap
1	50	564.68	[45]	565.02	0.06	566.80	0.38	567.98	0.59	566.82	0.38	568.86	0.74	566.80	0.38	**564.68**	0.00	565.99	0.23
2	50	611.53	[32]	662.84	8.39	615.66	0.68	619.35	1.28	612.75	0.20	617.48	0.97	620.15	1.41	**611.53**	0.00	614.23	0.44
3	50	618.04	[45]	664.73	7.56	620.78	0.44	629.59	1.87	**618.04**	0.00	620.50	0.40	632.48	2.34	**618.04**	0.00	618.04	0.00
4	75	798.53	[45]	857.84	7.43	801.60	0.38	809.13	1.33	808.84	1.29	817.71	2.40	803.32	0.60	**798.53**	0.00	803.51	0.62
5	75	839.62	[45]	949.98	13.14	**839.62**	0.00	858.98	2.31	**839.62**	0.00	858.95	2.30	842.50	0.34	**839.62**	0.00	841.63	0.24
6	75	930.64	[32]	1084.82	16.57	936.01	0.58	949.89	2.07	934.11	0.37	942.60	1.29	938.18	0.81	940.59	1.07	961.47	3.31
7	100	830.48	[45]	837.80	0.88	**830.48**	0.00	832.91	0.29	**830.48**	0.00	838.50	0.97	832.56	0.25	**830.48**	0.00	830.48	0.00
8	100	872.56	[32]	906.16	3.85	878.87	0.72	881.26	1.00	875.76	0.37	882.70	1.16	878.87	0.72	**872.56**	0.00	876.21	0.42
9	100	912.02	[32]	1000.27	9.68	942.31	3.32	955.95	4.82	912.64	0.07	921.97	1.09	980.42	7.50	914.23	0.24	918.45	0.71
10	150	1039.07	[45]	1076.88	3.64	1039.23	0.02	1052.65	1.31	1053.90	1.43	1074.38	3.40	1060.41	2.05	1046.71	0.74	1050.11	1.06
11	150	1093.37	a.	1170.17	7.02	1098.84	0.50	1107.47	1.29	1093.57	0.02	1108.88	1.42	1170.70	7.07	**1093.37**	0.00	1100.95	0.69
12	150	1152.32	a.	1217.01	5.61	1175.23	1.99	1184.58	2.80	1155.44	0.27	1166.59	1.24	1178.34	2.26	**1152.32**	0.00	1158.88	0.57
13	199	1287.18	[45]	1364.50	6.01	1288.46	0.10	1296.33	0.71	1320.21	2.57	1340.98	4.18	1288.46	0.10	1298.89	0.91	1305.83	1.45
14	199	1339.36	a.	1464.20	9.32	1371.42	2.39	1384.13	3.34	1351.54	0.91	1367.91	2.13	1372.52	2.48	**1339.36**	0.00	1354.04	1.10
15	199	1420.72	b.	1544.21	8.69	1459.55	2.73	1488.71	4.79	1436.78	1.13	1454.91	2.41	1470.21	3.48	1423.91	0.22	1437.52	1.18
16	120	1002.49	[45]	1064.89	6.22	**1002.49**	0.00	1003.00	0.05	1004.47	0.20	1007.26	0.48	1004.69	0.22	**1002.49**	0.00	1003.07	0.06
17	120	1026.20	[32]	1104.67	7.65	1042.35	1.57	1042.79	1.62	1026.88	0.07	1035.23	0.88	1042.35	1.57	1042.46	1.58	1042.61	1.60
18	120	1098.15	[32]	1202.00	9.46	1129.16	2.82	1141.94	3.99	1099.09	0.09	1110.13	1.09	1129.16	2.82	1113.07	1.36	1118.63	1.86
19	100	813.30	[32]	887.22	9.09	813.50	0.02	813.98	0.08	814.07	0.09	823.01	1.19	813.50	0.02	813.50	0.02	819.81	0.80
20	100	848.93	[45]	963.06	13.44	**848.93**	0.00	852.89	0.47	855.14	0.73	859.06	1.19	**848.93**	0.00	860.12	1.32	860.12	1.32
21	100	909.06	[45]	952.29	4.76	**909.06**	0.00	914.04	0.55	**909.06**	0.00	915.38	0.70	**909.06**	0.00	**909.06**	0.00	909.06	0.00
Average cost				1025.74				970.84				968.24		970.65				961.46	
Avg. gap above BKS					7.55%		0.89%		1.74%		0.48%		1.51%		1.74%		0.36%		0.84%
NBKS						5				4				2		12			

a. GRASP/VNS + EvPR

b. GRASP/VNS with PR (Post-optimization)

Table 5. Comparison of GRASP/VNS with EvPR against other approaches from the literature

	Chao [7]		Scheuerer [45]		Lin et al. [32]		GRASP/VNS + EvPR	
Problem number	Avg. Cost	Rank	Avg. Cost	Rank	Avg. Cost	Rank	Avg. Cost	Rank
1	565.02	1	567.98	3	568.86	4	565.99	2
2	662.84	4	619.35	3	617.48	2	614.23	1
3	664.73	4	629.59	3	620.50	2	618.04	1
4	857.84	4	809.13	2	817.71	3	803.51	1
5	949.98	4	858.98	3	858.95	2	841.63	1
6	1084.82	4	949.89	2	942.60	1	961.47	3
7	837.80	3	832.91	2	838.50	4	830.48	1
8	906.16	4	881.26	2	882.70	3	876.21	1
9	1000.27	4	955.95	3	921.97	2	918.45	1
10	1076.88	4	1052.65	2	1074.38	3	1050.11	1
11	1170.17	4	1107.47	2	1108.88	3	1100.95	1
12	1217.01	4	1184.58	3	1166.59	2	1158.88	1
13	1364.50	4	1296.33	1	1340.98	3	1305.83	2
14	1464.20	4	1384.13	3	1367.91	2	1354.04	1
15	1544.21	4	1488.71	3	1454.91	2	1437.52	1
16	1064.89	4	1003.00	1	1007.26	3	1003.07	2
17	1104.67	4	1042.79	3	1035.23	1	1042.61	2
18	1202.00	4	1141.94	3	1110.13	1	1118.63	2
19	887.22	4	813.98	1	823.01	3	819.81	2
20	963.06	4	852.89	1	859.06	2	860.12	3
21	952.29	4	914.04	2	915.38	3	909.06	1
Average Rank		3.81		2.29		2.43		1.48
Sum of Ranks		80		48		51		31
Squared Sum of Ranks		6400		2304		2601		961

Table 6
Data for the Friedman test on the ranking of the average performance of each metaheuristic

Problem				Caramia and Guerrriero		Best seed		Worst seed	
Number	n	BKS	Ref.	Cost	Gap	Cost	Gap	Cost	Gap
1	50	564.68	[45]	566.80	0.38	566.09	0.25	566.82	0.38
2	50	611.53	[32]	620.15	1.41	613.61	0.34	615.18	0.60
3	50	618.04	[45]	632.48	2.34	618.04	0.00	618.04	0.00
4	75	798.53	[45]	803.32	0.60	799.34	0.10	798.87	0.04
5	75	839.62	[45]	842.50	0.34	843.05	0.41	842.47	0.34
6	75	930.64	[32]	938.18	0.81	948.42	1.91	980.34	5.34
7	100	830.48	[45]	832.56	0.25	830.48	0.00	830.48	0.00
8	100	872.56	[32]	878.87	0.72	879.51	0.80	881.17	0.99
9	100	912.02	[32]	980.42	7.50	914.23	0.24	920.07	0.88
10	150	1039.07	[45]	1060.41	2.05	1051.70	1.22	1053.22	1.36
11	150	1093.37	a.	1170.70	7.07	1106.41	1.19	1099.05	0.52
12	150	1152.32	a.	1178.34	2.26	1152.32	0.00	1162.16	0.85
13	199	1287.18	[45]	1288.46	0.10	1309.69	1.75	1307.35	1.57
14	199	1339.36	a.	1372.52	2.48	1347.61	0.62	1361.63	1.66
15	199	1420.72	b.	1470.21	3.48	1424.31	0.25	1438.61	1.26
16	120	1002.49	[45]	1004.69	0.22	1002.82	0.03	1002.49	0.00
17	120	1026.20	[32]	1042.35	1.57	1042.53	1.59	1042.63	1.60
18	120	1098.15	[32]	1129.16	2.82	1113.07	1.36	1122.88	2.25
19	100	813.30	[32]	813.50	0.02	821.85	1.05	821.85	1.05
20	100	848.93	[45]	848.93	0.00	860.12	1.32	860.12	1.32
21	100	909.06	[45]	909.06	0.00	909.06	0.00	909.06	0.00
Avg. gap above BKS					1.74%		0.69%		1.05%
Number of times better than Caramia and Guerriero						13		13	

a. GRASP/VNS + EvPR

b. GRASP/VNS with PR (Post-optimization)

Table 7

Comparison of a single run of GRASP/VNS with EvPR and Caramia and Guerriero's heuristic

Problem		GRASP/VNS with EvPR		Scheuerer [45]		Lin et al. [32]	
Number	n	Avg. time	Scaled time (x2.15)	Avg. time	Speed-up	Avg. time	Speed-up
1	50	1.17	2.51	9.51	3.79	6.80	2.71
2	50	1.29	2.77	9.60	3.47	6.67	2.41
3	50	1.05	2.27	11.24	4.95	5.59	2.46
4	75	2.69	5.79	18.49	3.20	16.32	2.82
5	75	2.82	6.07	15.16	2.50	14.42	2.37
6	75	2.89	6.22	18.62	2.99	13.65	2.19
7	100	6.05	13.02	33.60	2.58	24.96	1.92
8	100	6.96	14.97	25.66	1.71	24.03	1.61
9	100	8.38	18.03	30.47	1.69	21.75	1.21
10	150	18.84	40.54	60.94	1.50	63.61	1.57
11	150	21.20	45.62	56.17	1.23	60.33	1.32
12	150	25.78	55.49	63.71	1.15	51.70	0.93
13	199	43.94	94.56	165.41	1.75	119.56	1.26
14	199	45.57	98.08	132.06	1.35	113.75	1.16
15	199	59.83	128.76	154.10	1.20	93.87	0.73
16	120	14.73	31.69	43.14	1.36	41.46	1.31
17	120	13.17	28.35	33.73	1.19	38.81	1.37
18	120	12.69	27.31	31.78	1.16	31.34	1.15
19	100	5.21	11.21	28.84	2.57	29.58	2.64
20	100	5.62	12.09	24.57	2.03	28.47	2.36
21	100	6.31	13.58	26.84	1.98	24.03	1.77
Average		14.58	31.38	47.32		39.56	
Geometric Mean					1.96		1.66

Table 8

Comparison of the running time of GRASP/VNS with EvPR and other published methods

5. Conclusions

In this chapter we proposed a hybrid metaheuristic based on GRASP, VNS and path relinking to solve the truck and trailer routing problem. The constructive phase and the path relinking operator are based on a route-first, cluster-second approach. During the search the proposed metaheuristic explores infeasible solutions while the VNS component plays the role of repairing operator and improving mechanism. The computational experiments on a set of 21 standard test instances from the literature unveils the accuracy of the proposed GRASP/VNS with path relinking and the contribution of path relinking to the quality of solutions. Moreover, the proposed hybrid metaheuristic outperforms all previous published methods, and exhibits a very small variability when comparing the results of a single run against the average results over several runs.

After exploring different hybridization alternatives for the path relinking component, GRASP/VNS with evolutionary path relinking emerged as the overall winner, achieving a small average gap to best-known solutions of 0.84%, finding 12 out of 21 best-known solutions, and improving 3 of them with a single set of parameters. The GRASP/VNS with path relinking as post-optimization mechanism variant is 32% faster than GRASP/VNS with evolutionary path relinking without a significant sacrifice on the quality of the results, achieving an average gap to best-known solution of 1.09%, and improving the best-known solution of one of the larger problems.

Further research directions include the development of lower bounds and exact methods to solve the TTRP, and the study of the multi-objective TTRP in which the fleet size (number of trucks and number of trailers), and the total distance are used as objective functions.

References

[1] J. E. Beasley. Route-first cluster-second methods for vehicle routing. *Omega*, 11:403–408, 1983.

[2] R. Bixby. Solving real-world linear programs: A decade and more of progress. *Operations Research*, 50(1):3–15, 2002.

[3] M. Boudia, M. Louly, and C. Prins. A reactive GRASP and path relinking for a combined production-distribution problem. *Computers & Operations Research*, 34(11):3402–3419, 2007.

[4] V. Campos, M. Laguna, and R. Martí. Context-independent scatter and tabu search for permutation problems. *INFORMS Journal on Computing*, 17(1):111–122, 2005.

[5] M. Caramia and F. Guerriero. A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society*, 61:1168–1180, 2010.

[6] M. Caramia and F. Guerriero. A milk collection problem with incompatibility constraints. *Interfaces*, 40(2):130–143, 2010.

[7] I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29:33–51, 2002.

[8] W. Conover. *Practical nonparametric statistics*. Wiley, 1998.

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

[10] J. Desrosiers, F. Soumis, and M. Desrochers. Routing with time windows by column generation. *Networks*, 14(4):545–565, 1984.

[11] J. J. Dongarra. Performance of various computers using standard linear equations software. Technical Report CS89-95., University of Tennessee, Computer Science, 2009.

[12] J. J. Dongarra, R. Wade, and P. McMahan. Linpack benchmark - Java version, 2010. http://www.netlib.org/benchmark/linpackjava/.

[13] M. Drexl. A branch-and-price algorithm for the truck-and-trailer routing problem. Technical report, Deutsche Post Endowed Chair of Optimization of Distribution Networks - RWTH Aachen University, 2006.

[14] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon. A GRASP X ELS approach for the capacitated location-routing problem. *Computers & Operations Research*, 37(11):1912–1923, 2010.

[15] M. Ehrgott. *Multicriteria Optimization*. Springer, 2000.

[16] A. El-Fallahi, C. Prins, and R. Wolfler-Calvo. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, 35:1725–1741, 2008.

[17] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[18] M. M. Flood. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

[19] S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms's behaviour: a case study on the CEC' 2005 special session on real parameter optimization. *Journal of Heuristics*, 15:617–644, 2009.

[20] M. Gendreau, J.-Y. Potvin, O. Bräysy, G. Hasle, and A. Løkketangen. Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 143–170. Springer, 2008.

[21] J. C. Gerdessen. Vehicle routing problem with trailers. *European Journal of Operational Research*, 93:135–147, 1996.

[22] B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

[23] F. Glover and M. Laguna. *Tabu search*. Kluwer, 1997.

[24] B. L. Golden, S. Raghavan, and E. Wasil. *The Vehicle routing problem: latest advances and new Challenges*. Springer, 2008.

[25] P. Hansen and N. Mladenovic. Variable neighbourhood search: principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

[26] H. Hashimoto and M. Yagiura. A path relinking approach with an adaptive mechanism to control parameters for the vehicle routing problem with time windows. In J. I. van Hemert and C. Cotta, editors, *EvoCOP*, volume 4972 of *Lecture Notes in Computer Science*, pages 254–265. Springer, 2008.

[27] S. C. Ho and M. Gendreau. Path relinking for the vehicle routing problem. *Journal of Heuristics*, 12:55–72, 2006.

[28] A. Hoff and A. Løkketangen. A tabu search approach for milk collection in western Norway using trucks and trailers. In *TRISTAN VI: The Sixth Triennial Symposium on Transportation Analysis*, Phuket, Tailand, June 10-15, 2007.

[29] M. Laguna and R. Marti. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11(1):44–52, 1999.

[30] G. Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics*, 54:811–819, 2007.

[31] S. Lin, V. Yu, and S. Chou. A note on the truck and trailer routing problem. *Expert Systems with Applications*, 37(1):899–903, 2010.

[32] S.-W. Lin, V. F. Yu, and S.-Y. Chou. Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, 36(5):1683–1692, 2009.

[33] I. Or. *Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking*. PhD thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL., 1976.

[34] S. N. Parragh, K. F. Doerner, R. F. Hartl, and X. Gandibleux. A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks*, 54(4):227–242, 2009.

[35] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31:1985–2002, 2004.

[36] C. Prins. A GRASP X Evolutionary Local Search hybrid for the vehicle routing problem. In F. B. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, pages 35–53. Springer, 2009.

[37] C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence.*, 22:916–928, 2009.

[38] C. Prins, C. Prodhon, and R. Wolfler-Calvo. 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, 2006.

[39] M. Reghioui, C. Prins, and N. Labadi. GRASP with path relinking for the capacitated arc routing problem with time windows. In M. Giacobini, A. Brabazon, S. Cagnoni, G. D. Caro, R. Drechsler, M. Farooq, A. Fink, E. Lutton, P. Machado, S. Minner, M. O'Neill, J. Romero, F. Rothlauf, G. Squillero, H. Takagi, S. Uyar, and S. Yang, editors, *EvoWorkshops*, volume 4448 of *Lecture Notes in Computer Science*, pages 722–731. Springer, 2007.

[40] M. G. C. Resende, R. Martí, M. Gallego, and A. Duarte. GRASP and path relinking for the maxmin diversity problem. *Computers & Operations Research*, 37(3):498–508, 2010.

[41] M. G. C. Resende and C. C. Ribeiro. GRASP with path-relinking: recent advances and applications. In T. Ibaraki, K. Nonobe, and M. Yagiura, editors, *Metaheuristics: progress as real problem solvers*, pages 29–63. Springer, 2005.

[42] M. G. C. Resende and C. C. Ribeiro. GRASP. In E. K. Burke and G. Kendall, editors, *Search methodologies*. Springer, 2 edition, 2010. To appear.

[43] M. G. C. Resende and C. C. Ribeiro. Greedy randomized adaptive search procedures: Advances and applications. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*. Springer, 2 edition, 2010. To appear.

[44] M. G. C. Resende and R. F. Werneck. A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10:59–88, 2004.

[45] S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33:894–909, 2006.

[46] T. Schiavinotto and T. Stutzle. A review of metrics on permutations for search landscape analysis. *Computers & Operations Research*, 34(10):3143–3153, 2007.

[47] F. Semet. A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research*, 61:45–65, 1995.

[48] F. Semet and E. Taillard. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research*, 41:469–488, 1993.

[49] K. Sörensen. Distance measures based on the edit distance for permutation-type representations. *Journal of Heuristics*, 13(1):35–47, 2006.

[50] K. Sörensen and M. Sevaux. 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, 2009.

[51] K. Sörensen and M. Sevaux. Distance-based path relinking for the vehicle routing problem. In *TRISTAN VII: Seventh Triennial Symposium on Transportation Analysis*, pages 744–747, Tromsø, Norway, June 20-25, 2010.

[52] W. Souffriau, P. Vansteenwegen, G. Vanden Berghe, and D. Van Oudheusden. A Path Relinking approach for the team orienteering problem. *Computers & Operations Research*, 37(11):1853–1859, 2010.

[53] P. Toth and D. Vigo. *The Vehicle Routing Problem*. SIAM, Philadelphia, 2002.

[54] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794, 2010.

Appendix A. Notation

The symbols used through the chapter are summarized in the following tables:

Symbol	Meaning
m_t	Number of availale trucks
m_r	Number of availale trailers
Q_t	Truck capacity
Q_r	Trailer capacity
$N = \{1, \ldots, n\}$	Set of customers
N_v	Subset of vehicle customers
N_t	Subset of truck customers
q_i	Customer demand
c_{ij}	Distance between nodes i and j

Table A.1
Notation for the definition of the TTRP

Symbol	Meaning
$T = (0, t_1, \ldots, t_i, \ldots, t_n, 0)$	Giant tour
$H = (X, U, W)$	Auxiliary graph for the tour splitting procedure
X	Set of nodes of the auxiliary graph
U	Set of arcs of the auxiliary graph
W	Weight of the arcs in the auxiliary graph
(i, j)	Arc of the auxiliary graph
w_{ij}	Cost of arc (i, j)
$R_{ij} = (0, t_i, \ldots, t_j, 0)$	Route serving from customer t_i to customer t_j of giant tour T
$c(R_{ij})$	Cost (total distance) of route R_{ij}
Q_{ij}	Total demand of route R_{ij}
$[l, m]$	State of the dynamic programming method used for the cost of vehicle routes with subtours
F_{lm}	Cost of state $[l, m]$
θ_{lkm}	Cost of subtour $ST = (t_l, t_k, \ldots, t_m, t_l)$
α_{ij}	Truck consumption of arc (i, j) of the auxiliary graph
β_{ij}	Trailer consumption of arc (i, j) of the auxiliary graph
$\Lambda = (\delta, \tau, \rho, \nu, \lambda)$	Label for the solution of the resource-constrained shortest path problem
δ	Distance of label Λ
τ	Truck consumption of label Λ
ρ	Trailer consumption of label Λ
ν	Father node of label Λ
λ	Father label of label Λ
$\mathcal{L}_i$	Set of labels of node i in the auxiliary graph
$\Gamma(i)$	Set of successors of node i in the auxiliary graph

Table A.2
Notation for the route-first, cluster-second procedure

Symbol	Meaning
S	TTRP solution
S_0	Initial solution of VNS
b	Number of pairs of customers exchanged in the perturbation procedure of VNS
$\Phi(S)$	Infeasibility of solution S
μ	Infeasibility threshold
$ut(S)$	Number of trucks used in solution S
$ur(S)$	Number of trailers used in solution S
$T(S)$	Giant tour of solution S
ES	Pool of elite solutions
$d(S, S')$	Distance between solutions S and S'
$d(\mathbf{ES}, S)$	Distance of solution S to the pool **ES**
$f'(S)$	Modified objective function for the ordering of the pool
S_0	Initial solution of the path relinking operator
S_f	Final solution of the path relinking operator
P, **RS**	Sets of solutions produced by path relinking

Table A.3
Notation of VNS and path relinking

Symbol	Meaning
ns	Number of GRASP iterations
κ	Cardinality of the restricted candidate list of GRASP
ni	Number of VNS iterations
μ_{max}	Maximum infeasibility threshold of VNS
b_{max}	Maximum number of pairs for the shaking of VNS
Δ	Minimum distance threshold in path relinking
$\|\mathbf{ES}\|$	Cardinality of the pool of elite solutions
γ	Frequency of evolutionary path relinking

Table A.4
Parameters of the hybrid metaheuristic

Appendix B. Solution of the restricted STTRPSD

The cost of vehicle routes with subtours is obtained by solving a restricted version of the STTRPSD (Villegas et al. [54]). Firstly, we recall the notation. The solution of the restricted STTRPSD is obtained with a dynamic programming method in which state states $[l, m]$ $(i \leq l \leq j; l : t_l \in N_v; l \leq m \leq j)$ represents the use of vehicle customer t_l as the root of a subtour ending at customer t_m. F_{lm} denotes the cost of state $[l, m]$, and θ_{lkm} is the cost of a subtour $ST = (t_l, t_k, \ldots, t_m, t_l)$ rooted at vehicle customer t_l and visiting customer t_k to t_m. It is possible to find the structure of the route and its cost using the following recurrence relation:

$$F_{lj} = \begin{cases} c_{0t_i} & \text{if } l = i \text{ and } m = i \\ \min\limits_{k<m:\sum_{u=k+1}^{m} q_{t_u} \leq Q_t} \{F_{lk} + \theta_{i,k+1,m}\} & \text{if } l = i \text{ and } i < m \leq j \\ \min \left\{ \min\limits_{k<m:\sum_{u=k+1}^{m} q_{t_u} \leq Q_t} \left\{F_{lk} + \theta_{l,k+1,m}\right\}, \min\limits_{\substack{h<l:t_h \in N_v, \\ k=l-1:\sum_{u=k+2}^{m} q_{t_u} \leq Q_t}} \left\{F_{hk} + c_{t_h,t_l} + \theta_{l,k+2,m}\right\} \right\} & \begin{array}{c}\text{if } i < l \leq j : t_l \in N_v, \\ \text{and } l \leq m \leq j\end{array} \end{cases}$$

Since states $[l, m]$, do not include the return from the last vehicle customer to the main depot, the cost of the route is finally calculated as $c(R_{ij}) = \min_{i \leq l \leq j : t_l \in N_v}\{F_{lj} + c_{t_l,0}\}$.

Following the same approach as Villegas et al. [54], the dynamic programming method for the STTRPSD can be represented by an auxiliary graph $G = (V, A, Z)$. The node set V is composed of the nodes representing states $[l, m]$, and a dummy node ω representing the return to the main depot.

The arc set A contains three types of arcs. Arcs $([l, k], [l, m]), i \leq l \leq j, t_l \in N_v; k < m \leq j$ represent a subtour that serves customers $(t_{k+1}, \ldots, t_m)$ rooted at vehicle customer t_l without moving the trailer that is already parked at vehicle customer t_l; the cost of these arcs is given by $\theta_{l,k+1,m}$. Arcs of the form $([h, k], [l, m]), i \leq h < l \leq j, t_h, t_l \in N_v; k = l-1 \leq j : \sum_{u=k+2}^{m} q_{t_u} \leq Q_t$ represent a subtour that serves customers $(t_{k+2}, \ldots, t_m)$ rooted at vehicle customer t_l coming from vehicle customer t_h after performing the subtour that end at customer t_k; the cost of these arcs is $c_{t_h,t_l} + \theta_{l,k+2,m}$. Finally, we have arcs of the form $([l, j], \omega), t_l \in N_v$ representing the return of the complete vehicle to the main depot after serving the last customer of the route in a subtour rooted at vehicle customer t_l. The cost of these arcs is $c_{t_l,0}$.

We obtain the structure of the route and its cost by finding in G the shortest path from state $[i, i]$ to node ω. As shown in Villegas et al. [54], this problem can be solved efficiently

Arc	Cost	Calculation	Value
$([5,5],[5,2])$	$\theta_{2,3,5} = c_{t_2,t_3} + c_{t_3,t_4} + c_{t_4,t_5} + c_{t_5,t_2}$	$c_{5,3} + c_{3,1} + c_{1,2} + c_{2,5} = 1.41 + 2.24 + 1.00 + 1.41$	6.06
$([5,2],[4,4])$	$c_{t_2,t_6} + \theta_{6,6,6}$	$c_{5,4} + 0 = 2.00 + 0$	2.00
$([4,4],0)$	$c_{t_6,0}$	$c_{4,0}$	2.24
Total			10.30

Table B.1
Example of the cost of the arcs of G for the STTRPSD

without generating explicitly the auxiliary graph G. A small example follows. The location of the customers and its type is given in Figure B.1, the length of each square side in the grid is equal to 1 and the distance between nodes is Euclidean.

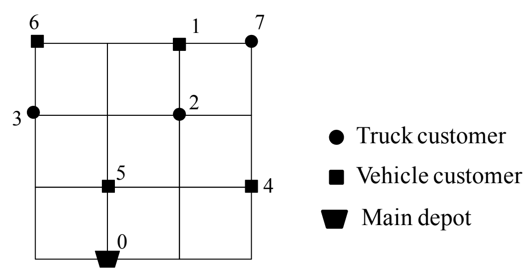


Fig. B.1. Data for the solution of the restricted STTRPSD

Given a giant tour $T = (0,6,5,3,1,2,4,7,0)$ for a TTRP with $Q_r = 2$, $Q_t = 3$ and unitary demands. We are interested in the structure and cost of route $R_{2,6} = (0,5,3,1,2,4,0)$. The auxiliary graph G is given in Figure B.2(a), the arcs in bold correspond to the shortest path, and the associated solution is given in Figure B.2(b). Note that in the auxiliary graph we replaced state $[l,m]$ with $[t_l,t_m]$ to simplify the presentation.

Table B.1 details the calculation of the cost of the arcs in the shortest path. Note that the cost of the route is 11.30, while the cost of the shortest path is 10.30 this is because it is necessary to add the cost of state $[i,i] = [2,2]$, that represents the departure from the main depot, in the recursion this is the first state with $F_{2,2} = c_{0,t_2} = c_{0,5} = 1.00$. Then, $c(R_{2,6}) = 11.30$.

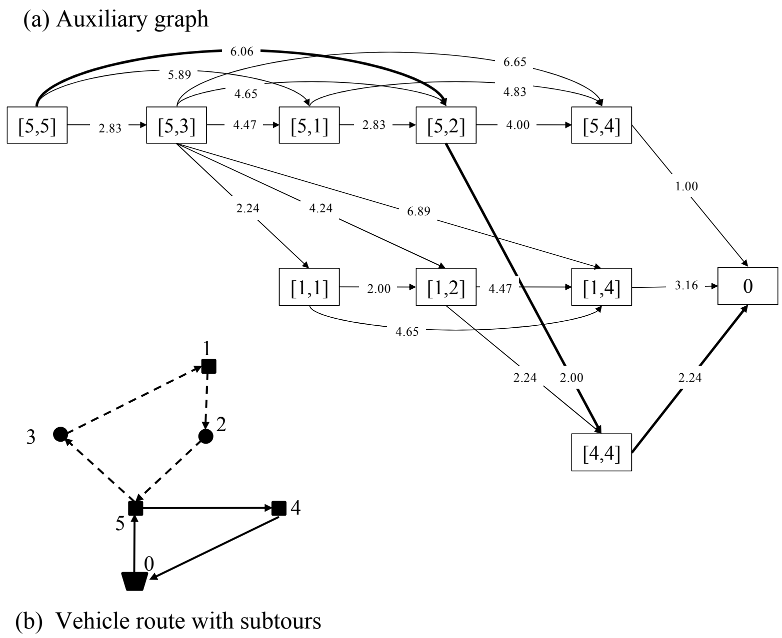


Fig. B.2. Example of the restricted STTRPSD used to find the cost of vehicle routes with subtours

This chapter has been accepted for publication in Computers & Operations Research and is available at:

J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. A GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*. Doi: 10.1016/j.cor.2010.11.011, 2010.

Chapter V

A matheuristic for the truck and trailer routing problem

A Matheuristic for the truck and trailer routing problem

Juan G. Villegas [a,b,c] Christian Prins [a] Caroline Prodhon [a] Andrés L. Medaglia [b]
Nubia Velasco [b]

[a] *Laboratoire d'Optimisation des Systèmes Industriels (LOSI), Institut Charles Delaunay, Université de Technologie de Troyes. BP 2060, 10010 Troyes Cedex, France*

[b] *Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes. A.A. 4976, Bogotá D.C., Colombia*

[c] *Departamento de Ingeniería Industrial, Universidad de Antioquia. A.A. 1226, Medellin, Colombia*

Abstract

We present a simple and effective matheuristic for the truck and trailer routing problem (TTRP) in which the routes of the local optima of a GRASP/VNS are used as columns of a set-partitioning formulation of the TTRP. This approach outperforms the previous state-of-the-art methods and improves the best-known solutions for several test instances from the literature.

Key words: Truck and trailer routing problem (TTRP), matheuristics, set-partitioning problem, greedy randomized adaptive search procedures (GRASP), variable neighborhood search (VNS).

1. Introduction

In the last years a new generation of hybrid optimization methods known as matheuristics has emerged. Matheuristics combine elements of exact mathematical programming algorithms and metaheuristics in a cooperative way [4,12,14]. Recently, Villegas et al. [18,20] presented a very effective GRASP/VNS with path relinking for the solution of the truck and trailer routing problem (TTRP). In their experiments a GRASP/VNS that uses path relinking as post-optimizer offered a good trade-off between solution quality and running time. This shows that GRASP/VNS is able to generate diverse high-quality solutions for the TTRP that can be used as input of a post-optimization phase. Therefore, in this chapter we solve the TTRP using a hybrid method that combines a GRASP/VNS

Email addresses: jg.villegas64@uniandes.edu.co, juan_guillermo.villegas@utt.fr, jvillega@udea.edu.co (Juan G. Villegas), christian.prins@utt.fr (Christian Prins), caroline.prodhon@utt.fr (Caroline Prodhon), amedagli@uniandes.edu.co (Andrés L. Medaglia), nvelasco@uniandes.edu.co (Nubia Velasco).

metaheuristic and a set-partitioning formulation of the problem.

The remainder of this chapter is organized as follows. Section 2 describes the TTRP and gives a brief literature review of the methods to solve it. Section 3 presents a set-partitioning formulation of the TTRP. Section 4 describes the elements of the proposed matheuristic. Computational results are presented in Section 5 followed by the conclusions in Section 6.

2. Problem definition and literature review

In the truck and trailer routing problem a heterogeneous fleet composed of m_t trucks and m_r trailers ($m_r < m_t$) serves a set of customers $N = \{1 \ldots, n\}$ from a main depot, denoted as node 0. Each customer $i \in N$ has a non-negative demand $q_i > 0$; the capacities of the trucks and the trailers are Q_t and Q_r, respectively; and the distance c_{ij} between any two points $i, j \in N \cup \{0\}$ is known. Some customers with limited maneuvering space or accessible through narrow roads must be served only by a truck, while other customers can be served either by a truck or by a complete vehicle (i.e., a truck pulling a trailer). These incompatibility constraints create a partition of N into two subsets: the subset of truck customers N_t accessible only by truck; and the subset of vehicle customers N_v accessible either by truck or by a complete vehicle. A distinguishing feature of the TTRP is that vehicle-customer locations can be used to park the trailer before serving truck customers. This possibility gives rise to complex routes with a main tour and one or more subtours.

The objective of the TTRP is to find a set of routes of minimum total distance such that: each customer is visited in a route performed by a compatible vehicle; the total demand of the customers visited in a route does not exceed the capacity of the allocated vehicle; and the number of required trucks and trailers is not greater than m_t and m_r, respectively.

Most of the methods to solve the TTRP follow the cluster-first, route-second approach. Chao [8] proposed a tabu search that solves a relaxed generalized assignment problem (RGAP) to allocate customers to routes in the clustering phase and uses an insertion heuristics to sequence the customers within the routes. Then, a multi-neighborhood search embedded within a hybrid tabu search/deterministic annealing method improves the initial solution. Likewise, Scheuerer [16] uses two constructive heuristics called T-Sweep and T-Cluster, that follow the cluster-first route-second principle, to find an initial solution that is improved with a tabu search. More recently, Caramia and Guerriero [7] have presented a mathematical programming-based heuristic that solves two subproblems sequentially. First, the customer-route assignment problem (CAP) assigns the customers to valid routes. Then, given the assignment of customers to routes, the route-definition problem (RDP) minimizes the tour length of each route.

In contrast, few researchers have tackled the TTRP with route-first, cluster-second methods. Lin et al. [13] developed a simple and effective simulated annealing that works on a permutation of the customers with additional dummy zeros to separate routes, along with a vector of binary variables of length $|N_v|$ representing the type of vehicle used to serve each customer in N_v. Whereas, Villegas et al. [18,20] solved the TTRP using a route-first, cluster-second procedure embedded within a hybrid metaheuristic based on a greedy randomized adaptive search procedure (GRASP), variable neighborhood search (VNS), and path relinking. Likewise, Villegas et al. [19] solved the single truck and trailer routing problem with satellite depots (STTRPSD) with a multi-start evolutionary local search and a hybrid GRASP/VND. These metaheuristics use a route-first cluster-second procedure and a VND as building blocks. For an updated review of the TTRP and other related problems the reader is referred to [13,18].

3. A set-partitioning formulation of the TTRP

Figure 1 illustrates the three types of routes in a TTRP solution. Let Ω be the set of feasible routes in the TTRP. Let $\Gamma \subseteq \Omega$ be the set of *pure truck routes* performed by trucks visiting customers in N_t and N_v; let $\Psi \subseteq \Omega$ be the set of *pure vehicle routes* performed by complete vehicles serving only customers in N_v; and finally let $\Pi \subseteq \Omega$ be the set of *vehicle routes with subtours*. The routes in the latter set Π involve a main tour performed by a complete vehicle visiting customers in N_v, and one or more subtours, in which the trailer is detached at a vehicle customer location followed by visits (only with the truck) to one or more customers in N_t and probably some customers in N_v. Let us define the binary parameter a_{ij} that takes the value of 1 if customer $i \in N$ is visited in pure truck route $j \in \Gamma$; it takes the value of 0, otherwise. Likewise, b_{ik} takes the value of 1 if customer $i \in N_v$ is visited in pure vehicle route $k \in \Psi$; it takes the value of 0, otherwise; and e_{il} takes the value of 1 if customer $i \in N$ is visited in vehicle route with subtours $l \in \Pi$, it takes the value of 0, otherwise.

Let us define the binary variables x_j that takes the value of 1 if and only if route $j \in \Gamma$ is used in the solution of the TTRP (it takes the value of 0, otherwise); y_k takes the value of 1 if and only if route $k \in \Psi$ is used (it takes the value of 0, otherwise); and z_l takes the value of 1 if and only if route $l \in \Pi$ is included in the TTRP solution (it takes the value of 0, otherwise). Let c_r be the cost (total distance) of any route $r \in \Omega(= \Gamma \cup \Psi \cup \Pi)$. The TTRP formulated as a set-partitioning problem (hereafter labeled as SPP) follows.

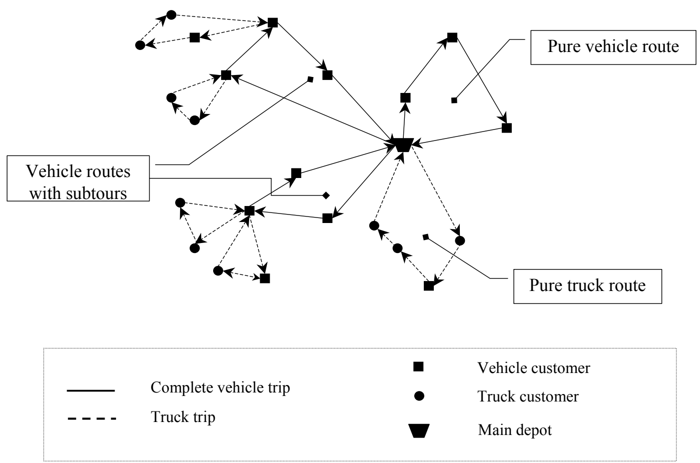


Fig. 1. Type of routes of a TTRP solution

$$\min \sum_{j=1}^{|\Gamma|} c_j x_j + \sum_{k=1}^{|\Psi|} c_k y_k + \sum_{l=1}^{|\Pi|} c_l z_l \tag{1}$$

subject to:

$$\sum_{j=1}^{|\Gamma|} a_{ij} x_j + \sum_{k=1}^{|\Psi|} b_{ik} y_k + \sum_{l=1}^{|\Pi|} e_{il} z_l = 1, \qquad \forall i \in N_v \tag{2}$$

$$\sum_{j=1}^{|\Gamma|} a_{ij} x_j + \sum_{l=1}^{|\Pi|} e_{il} z_l = 1, \qquad \forall i \in N_t \tag{3}$$

$$\sum_{j=1}^{|\Gamma|} x_j + \sum_{k=1}^{|\Psi|} y_k + \sum_{l=1}^{|\Pi|} z_l \leq m_t, \tag{4}$$

$$\sum_{k=1}^{|\Psi|} y_k + \sum_{l=1}^{|\Pi|} z_l \leq m_r, \tag{5}$$

$$x_j \in \{0,1\}, \qquad \forall j \in \Gamma \tag{6}$$

$$y_k \in \{0,1\}, \qquad \forall k \in \Psi \tag{7}$$

$$z_l \in \{0,1\}, \qquad \forall l \in \Pi \tag{8}$$

The objective function (1), to be minimized, is the total distance of the solution. Constraints (2) assure that each vehicle customer is visited exactly once; whereas, constraints (3) assure that each truck customer is visited exactly once by a truck route or a vehicle route with subtours. Constraints (4) and (5) impose an upper bound on the number of trucks and trailers that can be used, respectively. Since the number of feasible routes is huge, using this formulation directly is not possible. Therefore, we designed a heuristic solution approach that solves SPP over the subset of routes $\overline{\Omega} = \overline{\Gamma} \cup \overline{\Psi} \cup \overline{\Pi}$, where $\overline{\Gamma} \subseteq \Gamma$, $\overline{\Psi} \subseteq \Psi$ and $\overline{\Pi} \subseteq \Pi$.

4. Solution approach

The proposed heuristic follows a two-stage approach. In the first stage, a pool of columns (routes) $\overline{\Omega}$ is built by extracting the routes of the local optima visited during the search of a hybrid metaheuristic; then, in the second stage, we solve a SPP over the routes in $\overline{\Omega}$ to produce a possibly better solution. Different variants of this approach have been used before to solve several vehicle routing problems such as the capacitated vehicle routing problem [9,15], the vehicle routing problem with time windows [1], the split delivery vehicle routing problem [3], the heterogeneous fixed fleet vehicle routing problem [17], and a min-max selective vehicle routing problem [2]. In fact, one of the best solution methods for the capacitated vehicle routing problem is the adaptive memory programming approach of Rochat and Taillard [15] that uses a set-partitioning problem as post-optimizer.

4.1. *GRASP/VNS for the TTRP*

In this section, we briefly outline the main components of the hybrid GRASP/VNS, mechanism responsible of filling the pool of routes $\overline{\Omega}$ of the matheuristic approach. For a detailed description the reader is referred to [18].

4.1.1. *Greedy Randomized Construction*

The greedy randomized construction is performed by a tour splitting procedure (*Split*) that follows the route-first, cluster-second principle [5]. A giant tour T that visits all the customers in N is found using a randomized nearest neighbor heuristic with a restricted candidate list of size κ. Then, a solution S of the TTRP is derived from $T = (0, \ldots, t_i, \ldots, t_n, 0)$ by means of the tour splitting procedure. The tour splitting procedure constructs one auxiliary acyclic graph $H = (X, U, W)$, where the set of nodes contains a dummy node 0 and n nodes numbered 1 through n, and node i represents the customer in the i-th position of T. The arc set U contains arc $(i-1, j)$ if and only if the subsequence $(t_i, \ldots, t_j)$ can be served by a feasible route. Finally, the weight $w_{i-1,j}$ of arc $(i-1, j)$ is the total distance of the corresponding route. To derive S it is necessary to find the shortest path between 0 and n in H. The cost of the shortest path corresponds to the total distance of S and the arcs in the shortest path represent its routes.

The tour splitting procedure for the solution of the TTRP takes into account the heterogeneous fixed fleet by solving a resource-constrained shortest-path problem, where the resources are the available trucks and trailers. Moreover, if the arc $(i-1, j)$ represents a vehicle route with subtours its weight $w_{i-1,j}$ is found via a dynamic programming method that solves a restricted version of the single truck and trailer routing problem with satellite depots [19]; whereas for pure truck routes and pure vehicle routes it is easily calculated given their simple structure. However, it may be difficult to find feasible solutions with the tour splitting procedure in problems with a tight ratio between the total demand and the total capacity. Hence, if for a given giant tour there is no feasible solution (with at most m_t trucks and m_r trailers), an infeasible solution is obtained solving an unrestricted shortest-path problem.

4.1.2. *Variable neighborhood search (VNS)*

The VNS [11] procedure takes the incumbent solution S produced by the tour-splitting mechanism and tries to improve it by performing iteratively three steps: (i) randomly exchanges p pairs of customers from its giant tour T to obtain a new giant tour T'; (ii) derives a new solution S' by applying the tour splitting procedure to T'; and (iii) applies a variable neighborhood descent (VND) to S'. The VND explores sequentially the following five neighborhoods using a best-improvement strategy: Or-opt (in single routes and subtours), node exchange (in single routes/subtours and between pairs of routes/subtours), 2-opt (in single routes/subtours and between pairs of routes/subtours), node relocation (in single routes/subtours and between pairs of routes/subtours), and the root refining procedure for subtours described in [8]. The VNS is repeated over ni iterations, and the value of p is controlled dynamically between 1 and p_{max}. Since infeasible solutions are accepted as initial solutions and also during the search of the VNS, the incumbent solution of VNS is replaced by S' if $f(S') < f(S)$ and its infeasibility $\Phi(S') = \max\{0, \frac{ut(S')}{m_t} - 1\} + \max\{0, \frac{ur(S')}{m_r} - 1\}$ does not exceed a given limit μ, where $f(S)$ denotes the objective function of S, and $ut(S)$ and $ur(S)$ the number of trucks and trailers used in S. At each call of VNS, μ is initialized at μ_{max}, and it is updated at each iteration with $\mu \leftarrow \mu - \frac{\mu_{max}}{ni}$.

4.2. *Overview of the method*

Algorithm 1 presents the structure of the proposed matheuristic. The GRASP/VNS cycle (lines 2-31) is repeated ns times. At each main iteration, an initial solution for VNS is generated with the route-first cluster-second procedure (lines 3-4) followed by a VND improvement (line 5). The routes of this initial solution are added to the pool $\overline{\Omega}$ (line 7). Then, the VNS cycle (lines 13-30) improves the solution using a perturbation procedure (line 14) followed by VND (line 16). The pool is also updated with the routes of the local optimum reached after VND (line 17), even if it is infeasible. Each time a route is inserted in the pool we check that no duplicates are created. More complex domination rules could be implemented. However, the presolver of the optimizer will automatically reduce the size of the pool before solving the set-partitioning problem.

Within the VNS phase, the search alternates between solutions and giant tours; the perturbation is performed on the giant tour, while the VND operates over TTRP solutions, therefore procedure *Concat* (line 20) creates a giant tour from a solution by concatenating its routes in a single string.

After the GRASP/VNS main cycle, the post-optimization phase solves the SPP over the routes in the pool $\overline{\Omega}$ (line 32). The columns of the optimal solution are used to reconstruct a possibly better solution. Since a given column does not specify the structure of the route it represents (to build a solution from the set of optimal columns), it is necessary to record the route associated with each column in the pool.

5. Computational experiment

The matheuristic was implemented in Java and compiled using Eclipse JDT 3.5.1. For the solution of the set-partitioning problem we use Gurobi version 3.0 [10]. All the experiments were ran on a computer with an Intel Xeon running at 2.67 GHz (clock speed) under Windows 7 Enterprise Edition (64 bits) with 4 GB of RAM. For the sake of fairness, we used the same parameters as Villegas et al. [18] for the GRASP/VNS: $ns = 60$, $\kappa = 2$, $ni = 200$, $\mu_{max} = 0.25$, $p_{max} = 6$. We evaluated the performance of the proposed method on the test bed described in Chao [8], which comprises 21 problems ranging from 50 to 199 customers and three different fractions of truck customers for each size.

Table 1 shows the average and best results of the matheuristic over ten runs for each problem in the test bed. The gap to the previous best-known solution (BKS) is reported in the table. Negative gaps indicate that the proposed method improved the BKS; and values in bold indicate that the BKS has been found by the proposed method. Table 1 also includes the average running time in minutes.

In summary, the proposed matheuristic has a good performance for the solution of the TTRP, improving 7 out of 21 BKS. Remarkably, all these new BKS were found with a

Algorithm 1 Matheuristic for the TTRP

Parameters: ns, κ, ni, p_{max}, μ_{max}

Output: S^*

$f^* := \infty$, $\overline{\Omega} = \emptyset$
for $i = 1$ to ns **do**
  $T := RandomizedNearestNeighbor(N, \kappa)$
  $S := Split(T)$
  $S := VND(S)$
  $T := Concat(S)$
  Update $\overline{\Omega}$ with the routes of S
  if $f(S) < f^*$ and $\Phi(S) = 0$ **then**
    $S^* := S$
    $f^* := f(S)$
  end if
  $p := 1$, $\mu := \mu_{max}$
  for $j = 1$ to ni **do**
    $T' := perturb(T, p)$
    $S' := Split(T')$
    $S' := VND(S')$
    Update $\overline{\Omega}$ with the routes of S'
    if $f(S') < f(S)$ and $\Phi(S') \leq \mu$ **then**
      $S := S'$
      $T := Concat(S)$
    end if
    if $f(S') < f^*$ and $\Phi(S') = 0$ **then**
      $S^* := S'$
      $f^* := f(S')$
      $p := 1$
    else
      $p := \min\{p + 1, p_{max}\}$
    end if
    $\mu = \mu - \frac{\mu_{max}}{ni}$
  end for
end for
S^* :=Solve SPP over the pool $\overline{\Omega}$
return S^*

single set of parameters. On the contrary, most of the previous methods discovered the BKS during sensibility analysis with different sets of parameters or after extensive computational experiments [13,16]. Moreover, the difference between the average and best results of the proposed matheuristic is very narrow proving the robustness of the method.

Problem					Matheuristic				
Number	n	$\|Nt\|$	Previous BKS	New BKS	Best	Gap(%)	Avg.	Gap(%)	Time (min)
1	50	12	$564.68^{a,b,d,e}$	-	**564.68**	0.00	564.68	0.00	1.35
2	50	15	$611.53^{b,d,e}$	-	**611.53**	0.00	611.94	0.07	1.16
3	50	37	$618.04^{a,b,d,e}$	-	**618.04**	0.00	618.04	0.00	0.91
4	75	18	$798.53^{a,b,d,e}$	-	**798.53**	0.00	798.53	0.00	2.37
5	75	37	$839.62^{a,b,d,e}$	-	**839.62**	0.00	839.62	0.00	2.25
6	75	56	930.64^{b}	-	943.39	1.37	948.61	1.93	2.24
7	100	25	$830.48^{a,b,d,e}$	-	**830.48**	0.00	830.48	0.00	11.10
8	100	50	$872.56^{b,d}$	**870.94**	**870.94**	-0.19	872.42	-0.02	5.80
9	100	75	912.02^{b}	-	914.23	0.24	914.23	0.24	6.75
10	150	37	$1039.07^{a,b}$	**1036.96**	**1036.96**	-0.20	1038.37	-0.07	40.96
11	150	75	1093.37^{d}	**1091.91**	**1091.91**	-0.13	1092.64	-0.07	17.31
12	150	112	1152.32^{d}	**1149.41**	**1149.41**	-0.25	1150.26	-0.18	24.60
13	199	49	$1287.18^{a,b}$	**1284.71**	**1284.71**	-0.19	1288.50	0.10	177.72
14	199	99	1339.36^{d}	**1333.66**	**1333.66**	-0.43	1335.84	-0.26	22.55
15	199	149	1420.72^{d}	**1416.51**	**1416.51**	-0.30	1420.07	-0.05	28.95
16	120	30	$1002.49^{a,b,d}$	-	**1002.49**	0.00	1002.81	0.03	10.35
17	120	60	1026.20^{b}	-	1042.35	1.57	1042.57	1.60	9.93
18	120	90	1098.15^{b}	-	1112.68	1.32	1115.00	1.53	9.08
19	100	25	813.30^{b}	-	813.50	0.02	814.48	0.14	4.10
20	100	50	$848.93^{a,c}$	-	849.34	0.05	850.48	0.18	4.51
21	100	75	$909.06^{a,c,d,e}$	-	**909.06**	0.00	909.06	0.00	5.02

a. Solution found by Scheuerer [16]
b. Solution found by Lin et al. [13]
c. Solution found by Caramia and Guerriero [7]
d. Solution found by Villegas et al. [18]
e. Solution found by the matheuristic

Table 1
Results of the matheuristic on the test bed of Chao [8]

In terms of computing time, all but two instances have been solved in less than 30 minutes; being the exceptions problems 10 and 13, that take 40 minutes and almost 3 hours, respectively. It seems that problems with a smaller fraction of truck customers (problems 7, 10 and 13) are more difficult for the SPP post-optimization, note that within the same problem size these instances take in general longer running times. Moreover, for the same problem size the running time of the matheuristic can be highly variable when the fraction of truck customers varies.

Table 2 presents the information of the SPP solved in the post-optimization phase for each TTRP instance. The average size of the pool is given in column $|\overline{\Omega}|$. Whereas, column *# Cols. Eliminated (Presolve)* gives the average number of columns eliminated by the presolver of Gurobi. As can be seen in the last column of this table (*% Cols. eliminated*), the presolver eliminated a considerable number of columns, ranging from 31.6% to 62.6%, and with an average of 43.7%.

TTRP			Set-partitioning problem		
Instance	n	$\|Nt\|$	$\|\overline{\Omega}\|$	# Cols. Eliminated (Presolve)	% Cols. Eliminated
1	50	12	37054	16023	43.2
2	50	15	25668	13106	51.1
3	50	37	15031	9001	59.9
4	75	18	48102	24775	51.5
5	75	37	36230	20881	57.6
6	75	56	25390	15884	62.6
7	100	25	54409	21736	39.9
8	100	50	55199	22243	40.3
9	100	75	67183	22880	34.1
10	150	37	83910	30481	36.3
11	150	75	87475	29552	33.8
12	150	112	93777	29640	31.6
13	199	49	112374	38828	34.6
14	199	99	102381	37687	36.8
15	199	149	105555	36295	34.4
16	120	30	63165	24038	38.1
17	120	60	65307	22693	34.7
18	120	90	59754	21399	35.8
19	100	25	30440	16866	55.4
20	100	50	31384	17221	54.9
21	100	75	35147	18220	51.8
Average % of columns eliminated					43.7

Table 2
Size of the set-partitioning problems solved in the post-optimization phase

To evaluate the contribution of the post-optimization phase it is necessary to compare the results of the matheuristic with the results obtained with a simple GRASP/VNS and other methods from the literature. Table 3 presents the results of GRASP/VNS without any post-optimization phase, the GRASP/VNS with evolutionary path relinking (EvPR) proposed by Villegas et al. [18], and the simulated annealing (SA) of Lin et al. [13] (SA). Table 3 reports the best (*Best*) and average (*Avg.*) cost and the average time in minutes (*Time*) over ten runs for each problem and method. The last rows of the table summarize the average gaps above BKS, the number of times that each method found the best-known solution (NBKS) and the average running time. As it can be seen, the proposed matheuristic outperforms the previous methods from the literature, with a very small average gap to BKS of 0.33% that is almost one third of the gap of EvPR (the second best method). Values in bold indicate that the BKS has been found by a given method. The matheuristic approach found 15 out of 21 BKS compared to 8 found by EvPR, and 5 found by SA. Moreover, except for the BKS of problem 9, the proposed matheuristic also found the BKS retrieved by the other two methods.

The comparison with GRASP/VNS reveals the effectiveness of the post-optimization

Problem			GRASP/VNS			EvPR			SA			Matheuristic		
Number	n	BKS	Best	Avg.	Time[1]	Best	Avg.	Time[1]	Best	Avg.	Time[2]	Best	Avg.	Time[1]
1	50	564.68	**564.68**	568.31	0.91	**564.68**	565.99	1.17	566.82	568.86	6.8	**564.68**	564.68	1.35
2	50	611.53	614.27	617.53	1.00	**611.53**	614.23	1.29	612.75	617.48	6.67	**611.53**	611.94	1.16
3	50	618.04	**618.04**	619.07	0.86	**618.04**	618.04	1.05	**618.04**	620.50	5.59	**618.04**	618.04	0.91
4	75	798.53	802.41	815.16	1.86	**798.53**	803.51	2.69	808.84	817.71	16.32	**798.53**	798.53	2.37
5	75	839.62	841.81	857.79	1.99	**839.62**	841.63	2.82	**839.62**	858.95	14.42	**839.62**	839.62	2.25
6	75	930.64	989.71	1040.19	1.95	940.59	961.47	2.89	934.11	942.60	13.65	943.39	948.61	2.24
7	100	830.48	830.62	832.27	4.11	**830.48**	830.48	6.05	**830.48**	838.50	24.96	**830.48**	830.48	11.10
8	100	870.94	881.53	885.01	4.35	872.56	876.21	6.96	875.76	882.70	24.03	**870.94**	872.42	5.80
9	100	912.02	916.63	930.55	5.67	914.23	918.45	8.38	**912.64**	921.97	21.75	914.23	914.23	6.75
10	150	1036.96	1050.76	1062.03	9.95	1046.71	1050.11	18.84	1053.90	1074.38	63.61	**1036.96**	1038.37	40.96
11	150	1091.91	1114.64	1122.43	11.02	1093.37	1100.95	21.20	1093.57	1108.88	60.33	**1091.91**	1092.64	17.31
12	150	1149.41	1159.88	1174.89	14.00	1152.32	1158.88	25.78	1155.44	1166.59	51.7	**1149.41**	1150.26	24.60
13	199	1284.71	1319.38	1332.55	18.71	1298.89	1305.83	43.94	1320.21	1340.98	119.56	**1284.71**	1288.50	177.72
14	199	1333.66	1380.86	1395.50	20.07	1339.36	1354.04	45.57	1351.54	1367.91	113.75	**1333.66**	1335.84	22.55
15	199	1416.51	1454.10	1462.23	25.14	1423.91	1437.52	59.83	1436.78	1454.91	93.87	**1416.51**	1420.07	28.95
16	120	1002.49	1003.99	1005.88	8.13	**1002.49**	1003.07	14.73	1004.47	1007.26	41.46	**1002.49**	1002.81	10.35
17	120	1026.20	1045.08	1050.86	8.09	1042.46	1042.61	13.17	1026.88	1035.23	38.81	1042.35	1042.57	9.93
18	120	1098.15	1121.07	1128.51	7.73	1113.07	1118.63	12.69	1099.09	1110.13	31.34	1112.68	1115.00	9.08
19	100	813.30	817.11	820.94	3.82	813.50	819.81	5.21	814.07	823.01	29.58	813.50	814.48	4.10
20	100	848.93	860.12	861.34	4.21	860.12	860.12	5.62	855.14	859.06	28.47	849.34	850.48	4.51
21	100	909.06	912.35	913.62	4.59	**909.06**	909.06	6.31	**909.06**	915.38	24.03	**909.06**	909.06	5.02
Avg. gap above BKS			1.37%	2.34%		0.44%	0.92%		0.57%	1.59%		0.22%	0.33%	
NBKS			2			8			5			15		
Avg. time (min)					7.53			14.58			39.56			18.52

Table 3
Comparison of the matheuristic with other methods

[1]Average time in minutes on a 2.7 GHz Intel Xeon
[2]Average time in minutes on a 1.5 GHz Pentium IV

phase. Without such a phase, GRASP/VNS is not competitive, but the post-optimization phase increases the average running time by a factor of 1.49. Moreover, the average running time of the matheuristic (18.52 min.) is greater than the average running time of EvPR (14.58 min.). Nonetheless, this value is highly biased by the running time of problem 13 (177.72 min.). If this value is not taken into account the average running time decreases to 10.58 min. It is also important to note that the running time of the matheuristic is shorter than the running time of EvPR in 17 out of 21 problems. This is better illustrated in Table 4, in which the time factor of the two methods with respect to GRASP/VNS is given, where $TimeFactor_{Method} = \frac{Time_{Method}}{Time_{GRASP/VNS}}$. Following the arguments of Bixby [6], the last row of this table presents the geometric mean of the time factors as an estimate of the increase of the running time.

Problem	Time factor	
	EvPR	Matheuristic
1	1.29	1.49
2	1.28	1.16
3	1.23	1.06
4	1.45	1.27
5	1.42	1.13
6	1.49	1.15
7	1.47	2.70
8	1.60	1.33
9	1.48	1.19
19	1.37	1.07
20	1.33	1.07
21	1.38	1.09
16	1.81	1.27
17	1.63	1.23
18	1.64	1.17
10	1.89	4.12
11	1.92	1.57
12	1.84	1.76
13	2.35	9.50
14	2.27	1.12
15	2.38	1.15
Geometric mean	1.61	1.49

Table 4
Time factor of EvPR and the matheuristic with respect to GRASP/VNS

6. Conclusions and future work

In this work we presented a set-partitioning formulation of the truck and trailer routing problem that is used in a two-phase matheuristic. First, routes are picked from the local optima found by a GRASP/VNS. Second, these routes are used as columns of the set-partitioning formulation that is solved using a commercial mixed-integer programming optimizer. With this post-optimization approach the performance of GRASP/VNS is clearly improved giving rise to a very effective matheuristic combining GRASP and mathematical programming. The proposed matheuristic improved the best-known solutions for 7 out of 21 test problems and outperforms all the other previous methods from the literature. Nonetheless, there is still room for improvement for the proposed method, mainly to reduce and stabilize its running time. Future research directions include the use of specialized methods to solve the set-partitioning problem and the implementation of more elaborated pool-management strategies to reduce the number of columns of the set-partitioning problem.

References

[1] G. Alvarenga, G. Mateus, and G. Detomi. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6):1561–1584, 2007.

[2] C. Arbex Valle, L. Conegundes Martinez, A. Salles da Cunha, and G. R. Mateus. Heuristic and exact algorithms for a min-max selective vehicle routing problem. *Computers & Operations Research*, Oct. 2010.

[3] C. Archetti, M. G. Speranza, and M. W. P. Savelsbergh. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, 42(1):22–31, 2008.

[4] M. O. Ball. Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, 16(1):21–38, 2011.

[5] J. E. Beasley. Route-first cluster-second methods for vehicle routing. *Omega*, 11:403–408, 1983.

[6] R. Bixby. Solving real-world linear programs: A decade and more of progress. *Operations Research*, 50(1):3–15, 2002.

[7] M. Caramia and F. Guerriero. A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society*, 61:1168–1180, 2010.

[8] I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29:33–51, 2002.

[9] C. Groër, B. Golden, and E. Wasil. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, 2(2):79–101, 2010.

[10] Gurobi Optimization. Gurobi optimizer, Java reference manual, 2010. http://www.gurobi.com/doc/30/refman/node201.html.

[11] P. Hansen and N. Mladenovic. Variable neighbourhood search: principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

[12] L. Jourdan, M. Basseur, and E.-G. Talbi. Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, 199(3):620–629, 2009.

[13] S.-W. Lin, V. F. Yu, and S.-Y. Chou. Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, 36(5):1683–1692, 2009.

[14] J. Puchinger and G. R. Raidl. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In J. Mira and J. R. Álvarez, editors, *IWINAC*, volume 3562 of *Lecture Notes in Computer Science*, pages 41–53, 2005.

[15] Y. Rochat and E. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1):147–167, 1995.

[16] S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33:894–909, 2006.

[17] E. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO*, 33(1):1–14, 1999.

[18] J. G. Villegas, C. Prins, C. Prodhon, A. Medaglia, and N. Velasco. A GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 2010. Under review.

[19] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots.

Engineering Applications of Artificial Intelligence, 23(5):780–794, 2010.

[20] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND with path relinking for the truck and trailer routing problem. In *TRISTAN VII: Seventh Triennial Symposium on Transportation Analysis*, Tromsø, Norway, June 20-25 2010.

PART III: jSplit: A route-first cluster-second framework

Chapter VI

A route-first cluster-second computational framework for vehicle routing heuristics

A route-first cluster-second computational framework for vehicle routing heuristics

Juan G. Villegas[a,b,c] Christian Prins[a] Caroline Prodhon[a] Andrés L. Medaglia[b] Nubia Velasco[b]

[a]*Laboratoire d'Optimisation des Systèmes Industriels (LOSI), Institut Charles Delaunay, Université de Technologie de Troyes. BP 2060, 10010 Troyes Cedex, France*

[b]*Centro para la Optimización y Probabilidad Aplicada (COPA), Departamento de Ingeniería Industrial, Universidad de los Andes. A.A. 4976, Bogotá D.C., Colombia*

[c]*Departamento de Ingeniería Industrial, Universidad de Antioquia. A.A. 1226, Medellin, Colombia*

Abstract

When dealing with realistic situations, the vehicle routing problem (VRP) has important side constraints such as: mixed pick-ups and deliveries, heterogeneous vehicles, fixed fleet, and periodicity, among others. Thus, we developed an extensible computational object-oriented framework for rapid prototyping of heuristic methods to solve different VRP extensions. The framework provides a set of built-in components for the development of route-first cluster-second heuristics (a.k.a. split). This architecture allows the user to focus on the specific elements of the VRP variant under consideration, and to reuse the built-in components for the algorithmic logic of the split procedure. We illustrate the flexibility of the framework with the implementation of a simple evolutionary strategy to solve the capacitated VRP and the truck and trailer routing problem.

Key words: Vehicle routing problem, route-first cluster-second heuristics, object-oriented framework, metaheuristics

1. Introduction

In the vehicle routing problem (VRP) a set of vehicles of limited capacities based at a main depot serves the demand of a geographically scattered set of customers. The objective of the VRP is to find a set of routes of minimum total length such that each customer

Email addresses: `jg.villegas64@uniandes.edu.co`, `juan_guillermo.villegas@utt.fr`, `jvillega@udea.edu.co` (Juan G. Villegas), `christian.prins@utt.fr` (Christian Prins), `caroline.prodhon@utt.fr` (Caroline Prodhon), `amedagli@uniandes.edu.co` (Andrés L. Medaglia), `nvelasco@uniandes.edu.co` (Nubia Velasco).

is visited exactly once, all the routes begin and end at the main depot and the total demand of the customers visited in each route does not exceed the capacity of the allocated vehicle. For an introductory tutorial on the VRP the reader is referred to [39]. Formally, the VRP is modeled using a graph $G = (V, A)$ where V is the set of vertices and A the set of arcs. In the set of vertices $V = \{0, \ldots, n\}$, the main depot is represented by vertex 0, and every vertex $i \in V \backslash \{0\}$ represents a customer with nonnegative demand $q_i > 0$. Each arc (i, j) has a nonnegative traversing cost c_{ij} that represents the distance between i and j. An unlimited fleet of homogeneous vehicles with capacity Q is based at the main depot.

Not only the VRP has received considerable attention because by being an NP-Hard problem [43] it poses challenges to the research community, but also because it can be used to model and optimize routing operations in a wide variety of industries such as soft drink distribution [33], mail delivery [60], home health care services [21], milk collection [7], maintenance operations in public utilities [45] and private companies [66], waste collection [58], and school bus routing [13], among others. As a consequence, when dealing with realistic situations the classical VRP is extended to incorporate different side constraints such as time windows [66], periodicity [2], multiple depots [29], accessibility constraints [7], heterogeneous fleets and multiple trips [50], among others. Moreover, a given company may have to face routing decisions involving different characteristics [66], and it is likely that after some time of operation the routing system will need to incorporate new constraints or objectives. Thus, good vehicle routing methods are those that are not just accurate and fast, but also simple and flexible [14], even if this flexibility comes at the expense of solution quality [40]. Moreover, it seems that the most successful metaheuristics to solve vehicle routing problems are over-engineered to solve specific variants of the problem and include a lot of parameters and components [27,39], which jeopardize its flexibility and simplicity.

Even though some researchers have proposed frameworks to solve vehicle routing problems, many of these studies do not tackle directly the flexibility and simplicity issues. For instance, Caseau et al. [8] present a metaheuristic factory for vehicle routing problems that hybridizes a generic insertion heuristic with constraint programming to check the feasibility at each step of the method. They show how this hybrid approach obtains high-quality solutions for the VRP with time windows but do not illustrate the flexibility of their framework in other VRP variants. Du and Wu [15] exploit the flexibility of object-oriented programming to develop a prototypical decision support system for vehicle routing that integrates solution methods and databases. Their system uses an insertion heuristic to solve routing problems ranging from the traveling salesman problem (TSP) to a VRP with time windows and backhauls. However, their focus is on the decision support system and database design rather than on the solution methods. On the other hand, Potvin et al. [49] show how the object-oriented paradigm facilitates the modeling and solution of vehicle routing problems within ALTO, a computer system for the design of routing heuristics. With ALTO the user can develop his/her own heuristics based on general procedures that resemble some of the classical constructive and local search procedures for the solution of the VRP. The authors illustrate ALTO on a routing problem faced by a mailing company in Canada. However, they focused on the development of heuristic methods to solve VRPs with capacity, maximum distance, and time-window constraints, but other modeling constraints are not discussed.

Hasle and Kloster [30] have presented SPIDER, a commercial VRP Solver capable of solving rich VRPs (i.e., vehicle routing problems that incorporate many real-world constraints). SPIDER uses a variable neighborhood descent (VND) procedure in which several neighborhoods are explored trying to improve the current solution. Moreover, the VND procedure is embedded within an iterative control that combines the elements of very large-scale neighborhood search and iterated local search. Using SPIDER the authors report competitive results for several variants of the VRP. Specifically, when tackling the pick-up and delivery problem with time windows (PDPTW) SPIDER reaches solutions of the same quality of state-of-the-art methods, but due to its rich model and uniform algorithmic approach it is less computationally efficient. More recently, Groër et al. [28] have introduced a public library of local search metaheuristics for vehicle routing problems; the authors show how different metaheuristics based on simulated annealing, record-to-record and tabu search can be implemented with the library and how their library can be integrated with exact methods. A metaheuristic method developed as an example of the library provides results that are within one percent deviation of the best-known solutions for the VRP. However, aside from a brief discussion on the extension for the multi-depot VRP and the VRP with time windows, no other VRP variants have been tackled with the library.

In the same vein, metaheuristics based on route-first, cluster-second procedures had successfully solved different node- and arc-routing problems such as the classical capacitated VRP [51,52], the distance-constrained VRP [45], the VRP with time windows [35], the VRP with pick-ups and deliveries [46,63], the heterogeneous fixed fleet VRP [53], the fleet size and mix VRP [31], the capacitated location-routing problem [17], the split-delivery VRP [5], the multi-compartment VRP [20], the cumulative capacitated VRP [47], the multi-depot VRP [64], the multi-compartment VRP with stochastic demands [44], the two-dimensional loading VRP [19], the single truck and trailer routing problem with satellite depots [64], the truck and trailer routing problem [65], the dial-a-ride problem [38], the VRP with backhauls [16], the team orienteering problem [6]; some integrated production, inventory, and distribution problems [24,42]; the capacitated arc routing problem (CARP)

[36], the periodic CARP [11], the biobjective CARP [37], the stochastic CARP [22], the CARP with time windows [55], the split-delivery CARP [34], and the multi-depot CARP [32], among others. However, all these methods have been developed independently with a specific implementation for each problem, giving the highest priority to efficiency (over flexibility).

Due to the proven effectiveness of route-first, cluster-second based metaheuristics, the purpose of this chapter is to present `jSplit`, a flexible and extensible computational object-oriented framework for rapid prototyping and implementation of split-based heuristics and metaheuristics for different variants of the VRP . Consequently, the user of `jSplit` does not have to implement the common logic of the route-first, cluster second procedure, but just has to focus on the unique aspects of its specific VRP variant (i.e., objective function and/or constraints). Under this design principle, `jSplit` provides a simple and flexible framework for the development of solutions methods for a wide range of VRP variants.

The remainder of the chapter is organized as follows. Section 2 reviews the route-first, cluster-second concept. Section 3 describes the architecture of the split-based framework. Section 4 illustrates the use of the framework for the solution of the classical capacitated VRP and the truck and trailer routing problem. Finally, Section 5 presents some conclusions and possible extensions of the framework.

2. Overview of route-first cluster-second heuristics

In the 1980s Beasley [3] introduced route-first, cluster-second heuristics (RFCS) for the VRP; and very early this principle was used to develop heuristics for different routing problems such as the fleet size and mix VRP [26], and the fleet size and mix CARP [61]. But it was only twenty years later that Prins [51] unveiled its potential as a component of metaheuristics for routing problems. The fundamental idea is to take a giant tour T visiting all the customers and to break it into VRP feasible routes using a tour splitting procedure.

Recently, Duhamel et al. [18] provided a review of the state-of-the-art of tour splitting heuristics along with a new depth-first variant to handle efficiently complex VRPs. Moreover, Prins et al. [54] performed an extensive computational experiment of tour splitting heuristics for the CARP and the capacitated VRP (CVRP). In this section we recall the route-first, cluster-second approach.

Given a giant tour $T = (0, t_1, \ldots, t_i, \ldots, t_n, 0)$, where t_i represents the customer in the i-th position of the tour, the split procedure creates an auxiliary acyclic graph $H = (X, U, W)$. The set of nodes X contains a dummy node 0 and n nodes numbered 1 through n, where node i represents customer t_i (i.e., the customer in the i-th position of T); the arc set U contains arc $(i-1, j)$ if and only if route $R_{ij} = (0, t_i, \ldots, t_j, 0)$ is a feasible route; and the weight $w_{i-1,j}$ of the arc $(i-1, j)$ is the cost of R_{ij}. Once the auxiliary graph has been built, a VRP feasible solution S is derived from T by finding the shortest path between nodes 0 and n in H. The cost of this shortest path corresponds to the cost of S and the arcs in the shortest path represent its routes. Algorithm 1 summarizes the general route-first, cluster-second procedure.

Algorithm 1 General route-first cluster-second procedure

Input: Giant tour T
Output: VRP solution S

Create an empty graph H
for $i := 1$ to $n-1$ **do**
    for $j := i+1$ to n **do**
        if $\mathit{checkRoute}(R_{ij}) = \mathit{true}$ **then**
            $w_{i-1,j} := \mathit{calculateRouteCost}(R_{ij})$
            add arc $(i-1, j)$ to H with cost $w_{i-1,j}$
        end if
    end for
end for
Find the shortest path from 0 to n in H
$S :=$ create S using the arcs (routes) of the shortest path
return S

Note that the giant tour T visits all the customers of the VRP, ignoring the capacity of the vehicles and any other side constraint of the problem. The flexibility of route-first cluster-second heuristics and metaheuristics comes from the fact that the feasibility of route R_{ij} is checked only when adding the arc $(i-1, j)$ to the auxiliary graph. Moreover, since the cost of the route is represented by the arc's weight, almost any cost structure can be modeled. On the other hand, other problem specific constraints, such as limited (heterogeneous) fleets or capacitated depots can be handled in the solution of the shortest-path problem.

Finally, it is well known that the VRP can be modeled as a set-partitioning problem [1], where the rows represent the customers of the VRP and the columns represent feasible routes. Since the set of feasible routes is of exponential size, *petal heuristics* [23,56] solve the VRP by generating a subset of routes of good quality and solving the set-partitioning

Element(Type)	Responsibilities
`VRPPackage` (Package)	Manages the VRP information, contains data structures to store routes and solutions, and provides simple utility procedures like printing and solution checking
`SplitFitness` (Abstract class)	Splits the giant tour to create VRP solutions
`ConstraintHandler` (Concrete class)	Handles the evaluation of the feasibility of the routes associated with the arcs of the auxiliary graph
`CostCalculator` (Abstract class)	Calculates the cost of the routes associated with the arcs of the auxiliary graph
`SplitConstraint` (Abstract class)	Evaluates the feasibility of a route with respect to a given constraint
`SplitGraph` (Concrete class)	Stores the auxiliary graph H
`ShortestPathAlgorithm` (Abstract class)	Finds the shortest path between the first and the last node of the auxiliary graph

Table 1
`jSplit` components and responsibilities

problem over this restricted set of columns. Ryan et al. [57], show that route-first, cluster-second heuristics and petal heuristics share the same principle. In some sense, T serves as an oracle of columns for the set-partitioning formulation, and since the resulting matrix has a column-interval set-partitioning problem structure, it is totally unimodular [4]. Moreover, the set-partitioning problem with such a structure can be reformulated and solved using a shortest-path problem [57].

3. jSplit: A framework for route-first cluster-second heuristics

With `jSplit` the user concentrates on the development of the specific components of its vehicle routing problem and does not have to reimplement the inner logic of the split procedure. Since `jSplit` has been built using the object-oriented paradigm, the user can reuse existing modules (classes) or extend them to model more complex VRP variants. `jSplit` has two main components: the first one is `jSplit` itself, where the route-first, cluster-second procedure lies; and the second one is an auxiliary component called `VRPPackage`, responsible of managing the information required to model VRPs.

The classes of `jSplit` and their relationships are depicted in Figure 1; and Table 1 summarizes the main responsibilities of each element of `jSplit`. The core of the framework is the class `SplitFitness` that constructs VRP solutions from giant tours. This construction process is performed by method `split` in the following way. First, to build the auxiliary graph, the `GiantTour` is scanned systematically (lines 2-9 of Algorithm 1) checking the feasibility of each route with the `ConstraintHandler` (line 4 of Algorithm 1). The `ConstraintHandler` dynamically loads and evaluates all the constraints of the problem following the template of `SplitConstraint`. If the route associated with one arc passes the feasibility check, its cost is calculated using the `CostCalculator` (line 5 of Algorithm 1) and the corresponding arc is added to the `SplitGraph` (line 6 of Algorithm 1). Once the auxiliary graph is built, the shortest path from dummy node 0 to node n

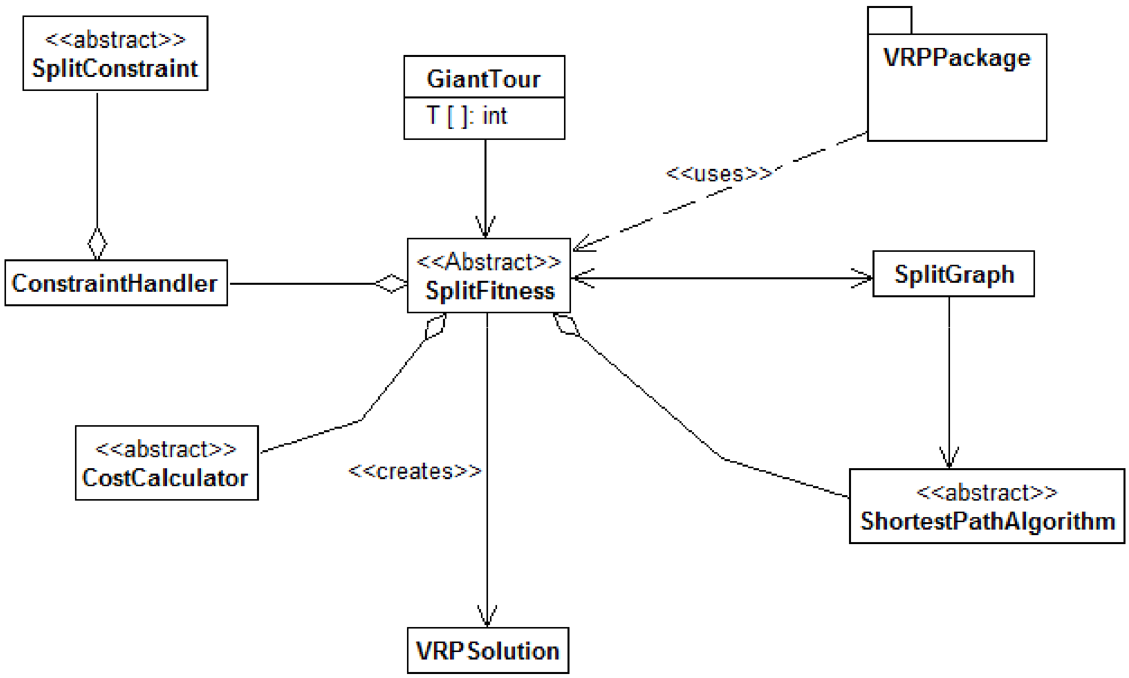


Fig. 1. `jSplit` architecture

is found by the `ShortestPathAlgorithm` implementation (line 10 of Algorithm 1). Then the solution is built using the routes corresponding to the arcs in the shortest path.

All the classes used by `SplitFitness` are loaded dynamically, making the process very flexible to the user who can incorporate the particular elements of his/her problem. For instance, capacity, route length, time windows, accessibility and backhauling can be verified using specific implementations of `SplitConstraint`, checking for feasibility while adding arcs to the auxiliary graph. Other structures like open routes, multiple depots and a heterogeneous fleet can be managed in the cost calculation of the arcs performed by the particular implementation of `CostCalculator`. The structure of many VRP variants can be handled using a simple shortest-path algorithm for directed acyclic graphs in the class `ShortestPathAlgorithm`. For problems with limited resources (e.g., multiple capacitated tb[]

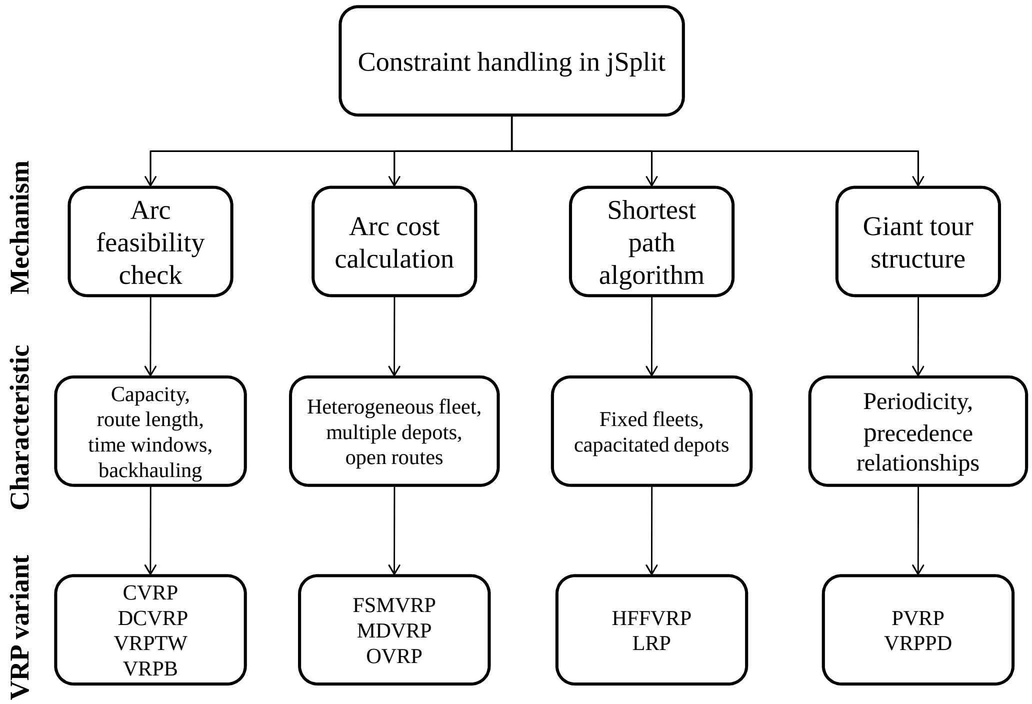


Fig. 2. Handling of different variants of the VRP within `jSplit`

depots or fixed fleet), other shortest-path algorithm implementations may be required. Finally, assignment structures and precedence relationships may be handled in the method that generates giant tours (e.g., by using a giant tour for each day in periodic VRPs). Figure 2 summarizes how different extensions of the VRP can be handled with `jSplit`.

4. jSplit illustrative examples

To illustrate its flexibility, in this section we provide two examples of vehicle routing problems tackled with `jSplit`. These examples are the classical capacitated VRP and the more complex truck and trailer routing problem. In both cases, we embedded `jSplit` within an evolutionary strategy (ES) that explores the giant-tour space. We chose a simple ES to avoid the use of many components that could obscure the contribution of the route-first cluster-second approach in the solution method.

Algorithm 2 presents the logic of the ES. The initial giant tour (T_0) is built using a simple nearest neighbor heuristic over the customer set V; then, the initial VRP solution

(S_0) is obtained using `jSplit`. Lines 6-8 initialize the best solution S^*, the best giant tour T^* and the best objective function f^*. After the initialization, the ES repeats its main cycle until a maximum time limit ($maxTime$) is reached. In the main cycle, the best giant tour T^* is mutated to obtain a new giant tour T' and a candidate solution S' is found using `jSplit`. When $f(S')$ improves f^* the ES updates T^*, S^* and f^*.

Algorithm 2 Simple evolutionary strategy for the solution of VRPs with `jSplit`

Parameters: $maxTime$

Output: VRP solution S^*

1: $f^* := \infty$
2: $S^* := \emptyset$
3: $runTime := 0$
4: $T_0 := NearestNeighborHeuristic(V)$
5: $S_0 := Split(T_0)$
6: $S^* := S_0$
7: $T^* := T_0$
8: $f^* := f(S_0)$
9: **do**
10: $T' := mutate(T^*)$
11: $S' := Split(T')$
12: **if** $f(S') < f^*$ **then**
13: $S^* := S'$
14: $T^* := T'$
15: $f^* := f(S')$
16: **end if**
17: $Update(runTime)$
18: **until** $runTime > maxTime$
19: **return** S^*

The mutation operator used in the ES randomly selects two positions of the giant tour and from a set of moves performs the one with the best objective function. The four moves considered are: (i) swap, (ii) forward insertion, (iii) backward insertion and, (iv) 2-opt. Figure 3 illustrates the four moves evaluated in the mutation operator.

4.1. *Capacitated VRP*

In `jSplit` the evaluation of the route-first, cluster-second procedure is parameterized with a configuration file that points to the specific files that contain the classes for route-cost calculation, constraint-feasibility check and shortest-path algorithm. Table 2 presents the classes used for the capacitated VRP. In this problem, the cost of a route is simply calculated as the total distance traversed by the vehicle, the constraints considered are

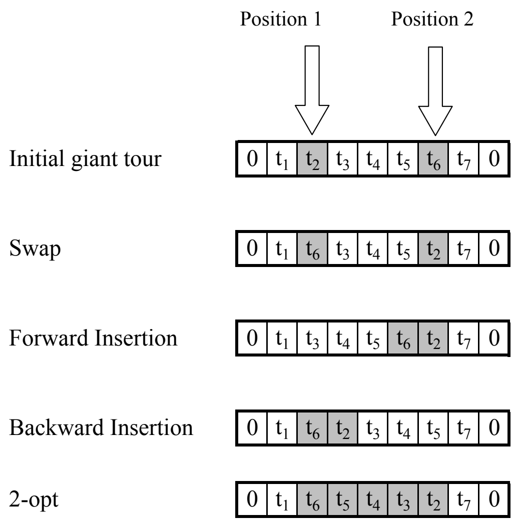


Fig. 3. Mutation operator of the ES

Class	Extended from class
`SimpleSplit`	`SplitFitness`
`SimpleCostCalculator`	`CostCalculator`
`SimpleCapacityConstraint`	`SplitConstraint`
`SimpleDurationConstraint`	`SplitConstraint`
`DAGBellmanAlgorithm`	`ShortestPathAlgorithm`

Table 2
Classes of `jSplit` for the CVRP

the classical capacity constraint and the total-route length constraint, and to solve the shortest-path problem a classical Bellman's algorithm for directed acyclic graphs is used.

To illustrate how constraints can be handled with `jSplit`, we describe the route-capacity check. The capacity constraint calculates the *load* of a vehicle in a route $R_{ij} = (0, t_i, \ldots, t_j, 0)$ and compares its value with the maximum capacity Q as follows:

$$load = \sum_{u=i}^{u=j} q_{t_u} \overset{?}{\leq} Q \tag{1}$$

In `jSplit` the capacity check is performed in the method `checkRoute` implemented by all classes extending the abstract class `SplitConstraint`. The logic of the this method of class `SimpleCapacityConstraint` is shown in Figure 4. Likewise, other constraints such as the maximum route-length can be easily handled by implementing similar methods that take fragments of the giant tour between positions i and j and check for the feasibility of the corresponding route.

```java
//Adds the demand of the customers of the route and compares it against the maximum capacity
public boolean checkRoute(int i, int j, int[] T){
    double load=0; //load of the vehicle
    for (int u = i; u <= j; u++){
        load=load+VRPInfo.getCustomer(T[u]).getDemand(dimension);
    }
    return (load<=maxCapacity);
}
```

Fig. 4. Method checkRoute of class SimpleCapacityConstraint

```java
// Calculates the length of a route serving from customer in position i to customer in position j
public double calculateRouteCost(int i, int j, int[] T){
    double length =0; //distance
    length =length+VRPInfo.getDistancesToDepots(T[i],0)+VRPInfo.getDistancesToDepots(T[j],0);
    for (int u = i; u < j; u++){
        length =length+VRPInfo.getDistancesPairCustomers(T[u],T[u+1]);
    }
    return length;
}
```

Fig. 5. Method calculateRouteCost of class SimpleCostCalculator

In a similar way, the route-cost calculation is performed by class `SimpleCostCalculator` using the method `calculateRouteCost` implemented by any class extending `CostCalculator`. In the CVRP the cost of a route R_{ij} is calculated adding the distance between the first customer and the depot, the distance from the last customer to the depot and the summation of the distances between consecutive customers in the giant tour between positions i and j . Figure 5 illustrates the `calculateRouteCost` method of class `SimpleCostCalculator`.

To evaluate the effectiveness of `jSplit` we tested our simple ES on the set of public VRP instances from Christofides et al. [10]. All the experiments were performed using a computer with an Intel Xeon processor running at 2.67 GHz under Windows 7 Enterprise Edition (64 bits) with 4 GB of RAM. The reported results come from a single run of the ES with two different stopping criteria (2 minutes for a *fast ES* and 10 minutes for a *slow ES*). Table 3 compares the results of the proposed ES with several classical heuristics for the CVRP. In this table, column BKS is the best known solution for each test problem. For each algorithm we report the cost of the obtained solution and the gap to BKS. CW refers to the results of the parallel savings heuristic of Clarke and Wright [12] followed by a 3-opt improvement, that has been implemented by Laporte and Semet [41]. Sweep refers to the heuristic of Gillet and Miller [25] implemented by Renaud et al. [56], as CW this method also includes a 3-opt improvement of each single route. Finally, Petal refers to the 1-Petal heuristic of Foster and Ryan [23] implemented by Renaud et al. [56] in which each petal is improved using a 4-opt* heuristic.

As shown in Table 3, the fast version of the ES obtains results that are comparable to those obtained by the very popular Clarke and Wright heuristic followed by a 3-opt improvement phase. The slow version of the ES decreases the average gap against the fast ES by roughly 1.4%. Moreover, it obtains results that are as good as those of the more

Problem	BKS	CW		Sweep		Petal		Fast ES (2 min)		Slow ES (10 min)	
		Cost	Gap (%)	Cost	Gap (%)	Cost	Gap (%)	Cost	Gap (%)	Cost	Gap (%)
1	524.61	578.56	10.28	531.90	1.39	531.90	1.39	583.80	11.28	583.80	11.28
2	835.26	888.04	6.32	884.20	5.96	885.02	5.96	882.05	5.60	882.05	5.60
3	826.14	878.70	6.36	846.34	1.23	836.34	1.23	865.88	4.81	865.88	4.81
4	1028.42	1128.24	9.71	1075.38	4.09	1070.50	4.09	1123.17	9.21	1048.93	1.99
5	1291.45	1386.84	7.39	1396.05	8.93	1406.84	8.93	1474.51	14.17	1416.59	9.69
6	555.43	616.66	11.02	560.08	0.84	560.08	0.84	570.55	2.72	570.55	2.72
7	909.68	974.79	7.16	965.51	6.51	968.89	6.51	957.03	5.20	957.03	5.20
8	865.94	968.73	11.87	883.56	1.37	877.80	1.37	925.07	6.83	925.07	6.83
9	1162.55	1284.63	10.50	1220.71	4.96	1220.20	4.96	1292.98	11.22	1263.52	8.69
10	1395.85	1521.94	9.03	1526.64	8.60	1515.95	8.60	1612.08	15.49	1534.12	9.91
11	1042.11	1048.53	0.62	1265.65	20.22	1252.84	20.22	1047.66	0.53	1047.30	0.50
12	819.56	824.42	0.59	919.51	0.64	824.77	0.64	829.20	1.18	829.20	1.18
13	1541.14	1587.93	3.04	1785.30	15.84	1773.69	15.09	1582.43	2.68	1579.47	2.49
14	866.37	868.50	0.25	911.81	5.24	894.77	3.28	881.44	1.74	881.44	1.74
Average gap			6.72		6.13		5.94		6.62		5.19

Table 3
Comparison of `jSplit` with classical heuristics for the CVRP

elaborated 1-Petal heuristic and better than those of the Sweep heuristic. It is noteworthy to see that all the classical heuristics have been designed to tackle the CVRP and include a local search phase to improve every single route, while the ES does not exploit any information of the problem and does not include a local search phase.

4.2. *Truck and Trailer Routing Problem*

To illustrate how `jSplit` can be used to solve more complex routing problems we also tested it to solve the truck and trailer routing problem (TTRP). The TTRP is an extension of the vehicle routing problem, where a heterogeneous fleet composed of m_t trucks and m_r trailers ($m_r < m_t$) is used to serve the set of customers. The capacities of the trucks and the trailers are Q_t and Q_r, respectively. The existence of accessibility constraints at some customers creates a partition of the customers into two subsets: the subset of truck customers V_t accessible only by truck; and the subset of vehicle customers V_v accessible either by truck or by a complete vehicle (i.e., a truck pulling a trailer). Due to the heterogeneity of the fleet and the accessibility constraints, a solution of the TTRP may have three types of routes (depicted in Figure 6): *pure truck routes* performed by a truck visiting customers in V_v and V_t; *pure vehicle routes* performed by a complete vehicle serving only customers in V_v; and finally *vehicle routes with subtours* performed by a complete vehicle. The latter type of route includes the case in which a trailer is detached at a vehicle customer in V_v to perform a subtour just with the truck visiting one or more customers in V_t (or even in V_v). The objective of the TTRP is to find a set of routes of minimum

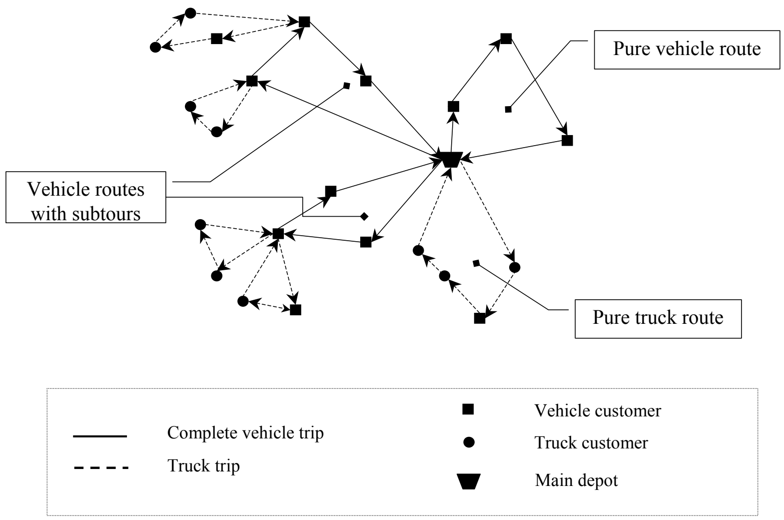


Fig. 6. Different types of routes in a TTRP

total distance such that: each customer is visited by a compatible vehicle exactly once; the total demand of the customers visited in a route or subtour does not exceed its capacity; and the number of required trucks and trailers is not greater than m_t and m_r, respectively.

To adapt the route-first, cluster-second approach for the solution of the TTRP it is necessary to take into account the accessibility constraints and the heterogeneous fixed fleet. We handle the accessibility constraints when checking the arc feasibility, and take into account the heterogeneous fixed fleet while solving the shortest path on H. Figure 7 illustrates how the different elements of the TTRP were tackled extending the different classes of `jSplit`.

A route will consume a different amount of resources depending on its type: pure truck routes only need trucks, while pure vehicle routes and vehicle routes with subtours require trucks and trailers. Therefore, it was necessary to extend `SplitGraphWithResources` from `SplitGraph` to include the resource consumption (trucks and trailers) in the arcs. The `SimpleCapacityConstraint` already included in the framework was extended to incorporate the heterogeneous fleet, but it was not necessary to reimplement the `checkRoute` method because the capacity check remains the same. In terms of capacity, a route will be infeasible if its load exceeds the maximum capacity of a complete vehicle, that is, the capacity of the truck plus the capacity of the trailer. The `TTRPCapacityConstraint` overrides the `initializeConstraint` method of the `SimpleCapacityConstraint` to change the maximum capacity value as shown in Figure 8.

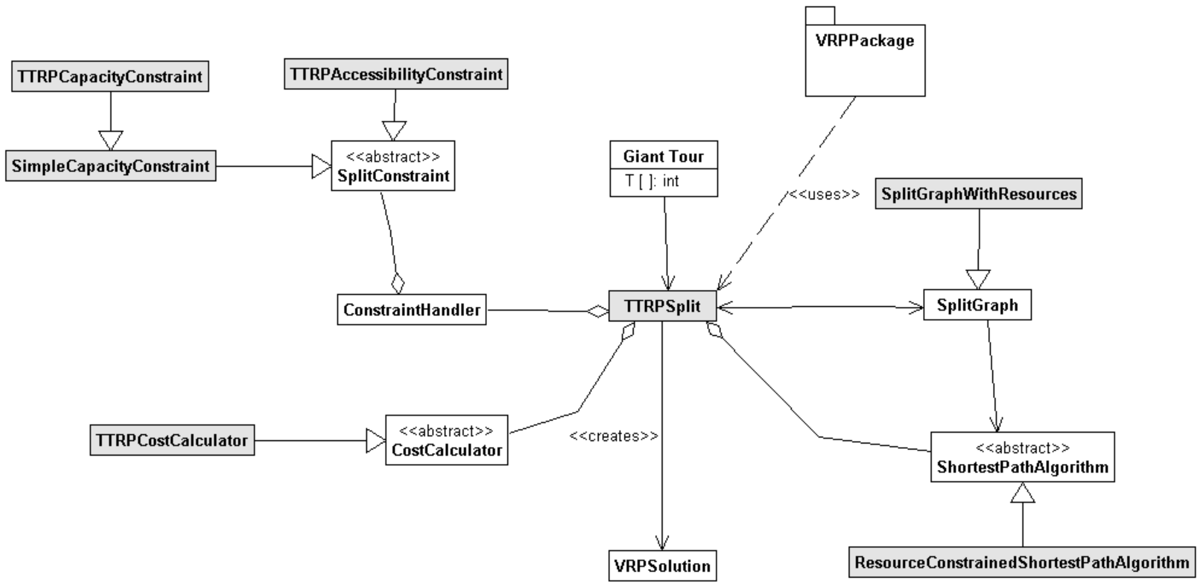


Fig. 7. Implementation of a route-first cluster-second method for the TTRP using `jSplit`

```
public class TTRPCapacityConstraint extends SimpleCapacityConstraint
{
    //The only difference with a simple capacity constraint is that the capacity to be
    //checked is the total capacity of the truck+the trailer
    public void initializeConstraint(){
        dimension = 0; //dimension of the demand of the customer that will be evaluated
        maxCapacity = VRPInfo.getVehicle(1).getCapacityValue(dimension)
                        +VRPInfo.getVehicle(2).getCapacityValue(dimension) ;
    }
}
```

Fig. 8. Class TTRPCapacityConstraint

To check the accessibility restriction we created `TTRPAccessibilityConstraint`. A route R_{ij} does not have accessibility problems if its load is less than Q_t. On the other hand, if the load is greater than Q_t the route will be infeasible with respect to the accessibility constraint if customer t_i is a truck customer. In that case the first customer of the route cannot be visited using the truck with the trailer that would depart from the main depot. The implementation of the logic of the method `checkRoute` of this constraint is shown in Figure 9.

The method for the cost calculation of a route depends on its type. If the arc represents a vehicle route with subtours, its cost is found by solving a restricted single truck and trailer routing problem with satellite depots (STTRPSD) [64]. On the other hand, the cost of

```java
//Verifies if the accessibility constraint of a route is met
public boolean checkRoute(int i, int j, int[] T){
    double load=0; //load of the vehicle
    for (int u = i; u <= j; u++){
        load=load+VRPInfo.getCustomer(T[u]).getDemand(dimension);
    }
    //There are no accessibility problems if the vehicle used is a truck
    if(load<=minCapacity)
    {
        return true;
    }
    else
    {
        TTRPCustomer firstCus=(TTRPCustomer)VRPInfo.getCustomer(T[i]);
        //If the capacity of the truck is exceeded (i.e. the trailer is used)
        //and the first customers is a truck customer the route is infeasible
        if(firstCus.isTruckCustomer())
        {
            return false;
        }
        //The route is feasible if the first customer is a vehicle customer
        else
        {
            return true;
        }
    }
}
```

Fig. 9. Method checkRoute of class TTRPAccessibilityConstraint

pure truck routes and pure vehicle routes is calculated with a procedure similar to that of the `SimpleCostCalculator`. This logic is implemented in `TTRPCostCalculator`. Finally, `ResourceConstrainedShortestPathAlgorithm` solves a resource-constrained shortest-path problem from node 0 to node n in H; and the routes of the arcs in the shortest path are used to create a TTRP solution.

To evaluate the effectiveness of `jSplit` in the solution of the TTRP we tested the simple ES on the 21-problem test bed introduced by Chao [9]. Table 4 summarizes the results of a single run of both versions of the ES for each problem. The last rows report the average gap to best-known solution and the number of feasible solutions found by each method. For comparison, Table 4 includes the results of the local search heuristics of Chao [9], that solves a relaxed generalized assignment problem and then applies a multi-neighborhood local search; the T-Sweep method of Scheuerer [59], that is an adaptation of the Sweep heuristic of Gillet an Miller [25]; and the T-Cluster method [59] that is a cluster-based insertion heuristic that constructs routes sequentially. Both T-Sweep and T-Cluster, include a steepest descent improvement procedure with three neighborhoods.

As shown in Table 4, the slow version of the ES outperforms specialized local search

		Scheuerer [59]				Chao [9]		*jSplit*			
Problem		T-Cluster		T-Sweep		Local Search		ES(10 min)		ES (2 min)	
Number	BKS	Cost	Gap (%)	Cost	Gap (%)	Cost	Gap (%)	Cost	Gap (%)	Cost	Gap (%)
1	564.68	651.87	15.44	644.80	14.19	646.02	14.41	639.64	13.28	639.64	13.28
2	611.53	697.51	14.06	722.46	18.14	739.90	20.99	687.49	12.42	687.49	12.42
3	618.04	766.25	23.98	797.65	29.06	774.78	25.36	705.41	14.14	705.41	14.14
4	798.53	979.79	22.70	Infeasible	-	943.47	18.15	915.96	14.71	984.40	23.28
5	839.62	1037.50	23.57	Infeasible	-	1130.85	34.69	1020.05	21.49	1020.05	21.49
6	930.64	1173.11	26.05	Infeasible	-	1236.69	32.89	Infeasible	-	Infeasible	-
7	830.48	904.77	8.95	901.14	8.51	906.31	9.13	872.11	5.01	974.54	17.35
8	870.94	965.90	10.90	1005.99	15.51	971.60	11.56	900.27	3.37	951.50	9.25
9	912.02	1081.21	18.55	1099.88	20.60	1106.66	21.34	1022.31	12.09	1035.76	13.57
10	1036.96	1167.38	12.58	1150.42	10.94	1159.78	11.84	1164.70	12.32	1242.11	19.78
11	1091.91	1274.67	16.74	1288.49	18.00	1288.74	18.03	1273.19	16.60	1347.99	23.45
12	1149.41	1438.11	25.12	1443.00	25.54	1453.82	26.48	1295.63	12.72	1393.67	21.25
13	1284.71	1485.67	15.64	1482.02	15.36	1481.40	15.31	1502.31	16.94	1527.18	18.87
14	1333.66	1611.99	20.87	1658.55	24.36	1624.96	21.84	1587.45	19.03	1632.32	22.39
15	1416.51	1748.31	23.42	1892.89	33.63	Infeasible	-	1712.75	20.91	1816.82	28.26
16	1002.49	1055.23	5.26	1383.57	38.01	1267.87	26.47	1047.82	4.52	1119.53	11.67
17	1026.20	1117.22	8.87	1416.14	38.00	1261.17	22.90	1103.80	7.56	1130.24	10.14
18	1098.15	1216.24	10.75	1614.11	46.98	1366.21	24.41	1236.45	12.59	1259.70	14.71
19	813.30	874.04	7.47	919.59	13.07	969.96	19.26	830.75	2.15	853.32	4.92
20	848.93	950.72	11.99	972.76	14.59	1140.47	34.34	887.51	4.55	901.93	6.24
21	909.06	1009.38	11.04	1096.08	20.57	1174.43	29.19	940.56	3.46	940.56	3.46
Average gap to BKS (%)			15.90		22.50		21.93		11.49		15.50
Feasible solutions		21		18		20		20		20	

Table 4

Comparison of the results of `jSplit` and local search heuristics for the TTRP

heuristics designed for the TTRP, obtaining an average gap to BKS that is half of the gap of the local search method of Chao and 5% better than the T-Cluster method. Even the fast version of the ES outperforms the T-Sweep method and the local search of Chao and is competitive with the T-Cluster method. In all, but one problem, the ES succeeded in obtaining feasible solutions. The only exception is problem 6, in which the demand-to-capacity ratio is very tight making this problem very difficult for the route-first, cluster-second approach.

5. Conclusions and future work

In this chapter we present `jSplit`, a general object-oriented framework for the development of route-first, cluster-second heuristics and metaheuristics for the solution of vehicle routing problems. Using an object-oriented architecture, the framework provides a set of built-in components for the development of route-first cluster-second heuristics. This architecture allows the user to focus on the elements of its specific variant of the VRP (mainly the feasibility check and the route-cost calculation). Thanks to the object-oriented architecture, the fact that no new heuristic rule is needed for each VRP variant and since the user has to concentrate only in the feasibility check and route-cost calculation, we believe that `jSplit` is part of the answer to the search for simple and flexible VRP methods. Moreover `jSplit` is publicly available at `http://copa.uniandes.edu.co`; and has been tested on two real-world VRP applications: the first application relates to the spare part distribution of an assembly plant in Bogotá [48] that was modeled as a VRP with fixed fleet; and the second application solves a crew scheduling problem for a midsize Colombian airline by means of a VRP with time windows [62].

Using the proposed framework we show how to model several VRP characteristics that include the basic capacity and route-length constraints as well as more complex structures such as heterogeneous fleets, accessibiity restrictions, limited availability of vehicles and non-trivial cost structures. Using a simple evolutionary strategy we show how `jSplit` can be embedded as a component of metaheuristics for the classical VRP and its extensions. When applied to the capacitated VRP and the more complex truck and trailer routing problem `jSplit` obtains results that are competitive with specialized constructive and local search heuristics.

Future directions of research include the development of general local search procedures that can be added to improve the quality of the solutions obtained with route-first cluster-second and the improvement of the running times of the splitting procedure.

References

[1] M. L. Balinski and R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2):300 – 304, 1964.

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

[3] J. E. Beasley. Route-first cluster-second methods for vehicle routing. *Omega*, 11:403–408, 1983.

[4] F. Boctor and J. Renaud. The column-circular, subsets-selection problem: complexity and solutions. *Computers & Operations Research*, 27(4):383–398, 2000.

[5] M. Boudia, C. Prins, and M. Reghioui. An effective memetic algorithm with population management for the split delivery vehicle routing problem. In T. Bartz-Beielstein, M. J. B. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 16–30, 2007.

[6] H. Bouly, D.-C. Dang, and A. Moukrim. A memetic algorithm for the team orienteering problem. *4OR-A Quarterly Journal of Operations Research*, 8(1):49–70, 2009.

[7] M. Caramia and F. Guerriero. A milk collection problem with incompatibility constraints. *Interfaces*, 40(2):130–143, 2010.

[8] Y. Caseau, F. Laburthe, and G. Silverstein. A meta-heuristic factory for vehicle routing problems. In J. Jaffar, editor, *CP*, volume 1713 of *Lecture Notes in Computer Science*, pages 144–158, 1999.

[9] I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29:33–51, 2002.

[10] N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial optimization*, pages 315–338. Wiley, 1979.

[11] F. Chu, N. Labadi, and C. Prins. A Scatter Search for the periodic capacitated arc routing problem. *European Journal of Operational Research*, 169(2):586–605, 2006.

[12] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivering points. *Operations Research*, 12:568–581, 1964.

[13] A. Corberán, E. Fernández, M. Laguna, and R. Martí. Heuristic solutions to the problem of routing school buses with multiple objectives. *Journal of the Operational Research Society*, 53:427–435, 2002.

[14] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53:512–522, 2002.

[15] T. C. Du and J.-L. Wu. Using object-oriented paradigm to develop an evolutional vehicle routing system. *Computers in Industry*, 44:229–249, 2001.

[16] C. Duhamel, P. Lacomme, and R. Phan. Approche de type GRASP X ELS pour le VRPB. In *ROADEF 2010*, Toulouse, February 24-26, 2010.

[17] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon. A GRASP X ELS approach for the capacitated location-routing problem. *Computers & Operations Research*, 37(11):1912–1923, 2010.

[18] C. Duhamel, P. Lacomme, and C. Prodhon. Efficient frameworks for greedy split and new depth first search split procedures for routing problems. *Computers & Operations Research*, page doi: 10.1016/j.cor.2010.09.010, 2010.

[19] C. Duhamel, P. Lacomme, A. Quilliot, and H. Toussaint. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research*, 38(3):617–640, 2011.

[20] A. El-Fallahi, C. Prins, and R. Wolfler-Calvo. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, 35:1725–1741, 2008.

[21] P. Eveborn, M. Ronnqvist, H. Einarsdottir, M. Eklund, K. Liden, and M. Almroth. Operations research improves quality and efficiency in home care. *Interfaces*, 39(1):18–34, 2009.

[22] G. Fleury, P. Lacomme, and C. Prins. Evolutionary algorithms for stochastic arc routing problems. In G. R. Raidl, S. Cagnoni, J. Branke, D. Corne, R. Drechsler, Y. Jin, C. G. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. D. Smith, and G. Squillero, editors, *EvoWorkshops*, volume 3005 of *Lecture Notes in Computer Science*, pages 501–512. Springer, 2004.

[23] B. A. Foster and D. M. Ryan. An integer programming approach to the vehicle scheduling problem. *Operational Research Quarterly*, 27(2):367 – 384, 1976.

[24] H. N. Geismar, G. Laporte, L. Lei, and C. Sriskandarajah. The integrated production and transportation scheduling problem for a product with a short lifespan. *INFORMS Journal on Computing*, 20(1):21–33, 2008.

[25] B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349, 1974.

[26] B. L. Golden, A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11:49–66, 1984.

[27] B. L. Golden, E. Wasil, J. P. Kelly, and I.-M. Chao. The impact of metaheuristics on solving the vehicle routing problem. In T. Crainic and G. Laporte, editors, *Fleet management and logistics*, pages 33–56. Springer, 1998.

[28] C. Groër, B. Golden, and E. Wasil. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, 2(2):79–101, 2010.

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

[30] G. Hasle and O. Kloster. Industrial vehicle routing. In G. Hasle, K.-A. Lie, and E. Quak, editors, *Geometric Modelling, Numerical Simulation, and Optimization*, pages 397–435. Springer, 2007.

[31] A. Imran, S. Salhi, and N. A. Wassan. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 2008.

[32] A. Kansou and A. Yassine. New upper bounds for the multi-depot capacitated arc routing problem. *International Journal of Metaheuristics*, 1(1):81–95, 2010.

[33] G. Kant, M. Jacks, and C. Aantjes. Coca-Cola enterprises optimizes vehicle routes for efficient product delivery. *Interfaces*, 38(1):40–50, 2008.

[34] N. Labadi, C. Prins, and M. Reghioui. An evolutionary algorithm with distance measure for the split delivery capacitated arc routing problem. In C. Cotta and J. I. van Hemert, editors, *Recent Advances in Evolutionary Computation for Combinatorial Optimization*, Studies in Computational Intelligence, pages 275–294. Springer, 2008.

[35] N. Labadi, C. Prins, and M. Reghioui. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO Operations Research*, 42:415–431, 2008.

[36] P. Lacomme, C. Prins, and W. Ramdane-Cherif. Competitive memetic algorithms for arc routing problems. *Annals of Operations Research*, 131(1-4):159–185, 2004.

[37] P. Lacomme, C. Prins, and M. Sevaux. A genetic algorithm for a bi-objective capacitated arc routing problem. *Computers & Operations Research*, 33(12):3473–3493, 2006.

[38] P. Lacomme, A. Quilliot, and X. Zhao. DARP: a tentative resolution using a multi-objective resolution scheme based on ELS. In *International Conference on Computers & Industrial Engineering, CIE 39.*, pages 1100–1105, Troyes, France, July 6-9 2009.

[39] G. Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics*, 54:811 – 819, 2007.

[40] G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

[41] G. Laporte and F. Semet. Classical heuristics for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 109–128. SIAM, 2002.

[42] L. Lei, S. Liu, A. Ruszczynski, and S. Park. On the integrated production, inventory, and distribution routing problem. *IIE Transactions*, 38(11):955–970, 2006.

[43] J. K. Lenstra and A. H. G. Rinnooy-Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–228, 1981.

[44] J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia, and N. Velasco. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research*, 37:1886–1898, 2009.

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

[46] G. Mosheiov. Vehicle routing with pick-up and delivery: tour-partitioning heuristics. *Computers & Industrial Engineering*, 34(3):669–684, 1998.

[47] S. U. Ngueveu, C. Prins, and R. Wolfler Calvo. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37(11):1877–1885, 2010.

[48] C. Pineda. Optimización al problema de entrega de repuestos a la red de concesionarios Sofasa - Bogotá. Master's thesis, Departamento de Ingeniería Industrial Universidad de los Andes, 2009. (In Spanish).

[49] J.-Y. Potvin, G. Lapalme, and J.-M. Rousseau. ALTO: a computer system for the design of vehicle routing algorithms. *Computers & Operations Research*, 16:451–470, 1989.

[50] C. Prins. Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case. *Journal of Mathematical Modelling and Algorithms*, 1:135–150, 2002.

[51] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31:1985–2002, 2004.

[52] C. Prins. A GRASP X Evolutionary Local Search hybrid for the vehicle routing problem. In F. B. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, pages 35–53. Springer, 2009.

[53] C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence.*, 22:916–928, 2009.

[54] C. Prins, N. Labadi, and M. Reghioui. Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research*, 47(2):507–535, 2009.

[55] M. Reghioui, C. Prins, and N. Labadi. GRASP with path relinking for the capacitated arc routing problem with time windows. In M. Giacobini, A. Brabazon, S. Cagnoni, G. D. Caro, R. Drechsler, M. Farooq, A. Fink, E. Lutton, P. Machado, S. Minner, M. O'Neill, J. Romero, F. Rothlauf, G. Squillero, H. Takagi, S. Uyar, and S. Yang, editors, *EvoWorkshops*, volume 4448 of *Lecture Notes in Computer Science*, pages 722–731. Springer, 2007.

[56] J. Renaud, F. F. Boctor, and G. Laporte. An improved petal heuristic for the vehicle routeing problem. *The Journal of the Operational Research Society*, 47(2):329 – 336, 1996.

[57] D. M. Ryan, C. Hjorring, and F. Glover. Extensions of the petal method for vehicle routeing. *The Journal of the Operational Research Society*, 44(3):289 – 296, 1993.

[58] S. Sahoo, S. Kim, B. Kim, B. Kraas, and A. Popov. Routing optimization for waste management. *Interfaces*, 35:24–36, 2005.

[59] S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33:894–909, 2006.

[60] C. D. Tarantilis, C. T. Kiranoudis, and N. C. Markatos. Use of the BATA algorithm and MIS to solve the mail carrier problem. *Applied Mathematical Modelling*, 26:481–500, 2002.

[61] G. Ulusoy. The fleet size and mix problem for capacitated arc routing. *European Journal of Operational Research*, 22(3):329–337, 1985.

[62] M. A. Vargas, J. L. Walteros, and A. L. Medaglia. Hybrid approach for airline crew pairing optimization for small carriers. In *IERC 2009: Industrial Engineering Research Conference*, Miami (US), May 30 - June 3, 2009.

[63] N. Velasco, P. Castagliola, P. Dejax, C. Guéret, and C. Prins. A memetic algorithm for a pick-up and delivery problem by helicopter. In F. B. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, volume 161 of *Studies in Computational Intelligence*, pages 173–190. Springer, 2009.

[64] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794, 2010.

[65] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND with path relinking for the truck and trailer routing problem. In *TRISTAN VII: Seventh Triennial Symposium on Transportation Analysis*, Tromsø, Norway, June 20-25 2010.

[66] D. Weigel and B. Cao. Applying GIS and OR techniques to solve Sears technician-dispatching and home delivery problems. *Interfaces*, 29(1):112–130, 1999.

General conclusion and Appendices

Chapter VII

General conclusion

General conclusion

Vehicle routing problems are a large family of combinatorial optimization problems with application in many different domains ranging from the distribution of goods to the delivery of services. The study of these problems has been motivated not only because of their computational complexity but also because the important savings that can be achieved in practical applications through the use of optimization techniques. Moreover, the current need to reduce green-house emissions gives even new reasons to design and implement efficient routing solutions.

In this thesis we have studied two vehicle routing problems with trailers, namely, the single truck and trailer routing problem with satellite depots (STTRPSD) and the truck and trailer routing problem (TTRP). In this type of routing problems the capacity of the vehicles has been increased with the use of detachable trailers. Even though this is a cost-effective way to increase the capacity of the fleet, at the same time it creates accessibility problems at some customers reachable only by narrow roads or with limited maneuvering space. Applications of vehicle routing problems with trailers can be found in distribution and collection operations in rural areas and crowded cities. For instance, in several European countries milk collection operations use truck with detachable tankers.

The STTRPSD models the case in which a single truck with a detachable trailer based at a main depot has to serve the demand of a set of customers accessible only by truck. This problem has an underlying location-routing structure because the trailer must be detached at designated parking places before visiting the customers with accessibility constraints. Likewise, a multi-depot vehicle routing problem (MDVRP) substructure is also present in the design of the routes that serve the customers with the truck.

For the solution of the STTRPSD we have developed several heuristics, metaheuristics and exact methods. The proposed heuristics include a cluster-first, route-second approach, an iterated route-first cluster-second procedure and a variable neighborhood descent (VND). Two metaheuristics based on GRASP (greedy randomized adaptive search procedures) and evolutionary local search have been implemented. Finally, we developed a branch-and-cut algorithm for the STTRPSD.

Computational experiments on a test bed of 32 randomly generated instances unveiled the effectiveness of a multi-start evolutionary local search (MS-ELS). Using a very simple scheme, this metaheuristic provides very good results in roughly one third of the time spent by a hybrid GRASP/VND. Moreover, for problems with up to 50 customers and 10 parking places, and for clustered problems with up to 100 customers and 20 parking places the optimality of the best solutions found by MS-ELS has been proved with the branch-and-cut algorithm. We also tested the MS-ELS on the MDVRP achieving competitive results with respect to state-of-the-art methods for this problem.

The branch-and-cut algorithm also proved to be effective in the solution of the STTRPSD. It solves optimally problems with up to 50 customers and 10 parking places in less than 15 minutes. Moreover, in clustered problems this limit increases up to 100 customers and 20 parking places, if the running time is increased to 3 hours.

While the STTRPSD models the single-vehicle case, the TTRP models the multi-vehicle case in which a heterogeneous fixed fleet of trucks and trailers is used to serve the demand of a set of customers, some of them with accessibility restrictions. Additional complicating elements of the TTRP are the fixed heterogeneous fleet (trucks and trucks with trailers) and three different types of routes that can be found in feasible solutions. Moreover, the TTRP generalizes in some sense the STTRPSD because some of the feasible routes have a STTRPSD-like structure. For the solution of the TTRP we have proposed a hybrid metaheuristic and a matheuristic.

The hybrid metaheuristic combines the elements of GRASP and path relinking. The greedy randomized construction of the hybrid metaheuristic is a route-first cluster-second procedure; while the improvement phase is performed by a VNS (variable neighborhood search). A path relinking procedure has been added to improve the results of the hybrid GRASP/VNS. Following a modular design, path relinking is used as a post-optimization procedure, as an intensification mechanism, or in a novel approach called evolutionary path relinking (EvPR).

On the other hand, the matheuristic combines metaheuristics and exact approaches in a two-phase method. In the first phase, a pool of routes is generated with the information of the local optima found by GRASP/VNS; then, a second phase solves a set-partitioning formulation of the TTRP (over the routes in the pool) to find a better solution.

Computational experiments on TTRP instances from the literature showed the good performance of the two proposed methods. Noteworthy is the contribution of the path relinking procedure to the quality of the results obtained by GRASP with path relinking. The use of path relinking as post-optimization procedure halves the average gap to best known solutions of GRASP/VNS with an increase in the running time of only 30%. Moreover, the best results of GRASP with path relinking were obtained by the evolutionary path relinking variant with an average gap to best-known solutions of less than 1%, but at the price of longer running times.

The results obtained with the matheuristic are better than those of GRASP with evolutionary path relinking and with comparable running times. The matheuristic improved 7 out of 21 best-known solutions and achieved an average gap to best known solutions of only 0.25%. In summary, both of the methods proposed in this thesis for the solution of the TTRP outperform the previous methods from the literature.

Motivated by the good results achieved with route-first, cluster-second methods developed for the STTRPSD and the TTRP, we designed and implemented an object-oriented framework for rapid prototyping of heuristic methods based on this principle. This framework provides the user with a set of reusable components that can be adapted to tackle different VRP extensions. In one of the examples, a simple evolutionary strategy (without any local search procedure), that uses the framework for the calculation of the objective function, provides results of comparable quality of those of constructive and local search heuristics specially designed for the TTRP.

In spite of the good results achieved on the STTRPSD and the TTRP, there is still room for improvement of some of the methods presented in this thesis. Currently we are working on new valid inequalities to be included on the branch-and-cut algorithm for the STTRPSD and on strategies to reduce the running time of the matheuristic for the TTRP.

It could be interesting to extend the methods proposed in this thesis to tackle the two-echelon vehicle routing problem, for which the STTRPSD is a special case. Likewise, given the good results of the matheuristic for the TTRP, exact approaches based on column generation using the set-partitioning formulation could be explored.

Finally, as has been done for other vehicle routing problems, natural extensions like the TTRP with time windows, multiple depots or heterogeneous trucks and trailers could be addressed in the future. Given that a non-trivial tradeoff between the fleet composition and the total distance exists, another promising avenue of research is the study of the multi-objective TTRP.

Appendix A

Publication summary

Publications summary

Journal articles

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794, 2010.

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*. Doi: 10.1016/j.cor.2010.11.011, 2010.

Conference proceedings

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND with path relinking for the truck and trailer routing problem. In *TRISTAN VII: Seventh Triennial Symposium on Transportation Analysis*, Tromsø (Norway), June 20-25, 2010.

– J.G. Villegas, A.L. Medaglia, C. Prins, C. Prodhon, and N. Velasco. GRASP/evolutionary local search hybrids for a truck and trailer routing problem. In *MIC 2009: The VIII Metaheuristics International Conference*, Hamburg (Germany), July 13-16, 2009.

– J. G. Villegas, A. L. Medaglia, J. E. Mendoza, C. Prins, C. Prodhon, and N. Velasco. A split-based framework for the vehicle routing problem. In *CLAIO 2008: XIV Congreso Latino Ibero Americano de Investigación de Operaciones*, Cartagena (Colombia), September 9-12, 2008.

Conference talks

– J. M. Belenguer, E. Benavent, A. Martínez, C. Prins, C. Prodhon, and J.G. Villegas. A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots. In *SEIO 2010: XXXII Congreso Nacional de Estadística e Investigación Operativa*, A Coruña (Spain), September 14-17, 2010.

– J.G. Villegas, J. M. Belenguer, E. Benavent, A. Martínez, C. Prins, and C. Prodhon. A cutting plane approach for the single truck and trailer routing problem with satellite depots. In *EURO 2010: XXIV European Conference on Operational Research*, Lisbon (Portugal), July 11-14, 2010.

– J. G. Villegas, A. L. Medaglia, C. Prins, C. Prodhon, and N. Velasco. Solving the truck and trailer routing problem with a route-first, cluster-second framework. In *ALIO/INFORMS Joint International Meeting*, Buenos Aires (Argentina), June 6-9, 2010.

– J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia and N. Velasco. Metaheuristics for a truck and trailer routing problem. In *CORAL 2009: III Combinatorial Optimization, Routing and Location Meeting*, Elche (Spain), June 10-12, 2009.

– J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia and N. Velasco, Vehicle routing with trailers. In *4me Forum des doctorants de l'UTT*, Troyes (France), May 13, 2009.

– J. G. Villegas, N. Velasco, C. Prins, J. E. Mendoza, and A. L. Medaglia. A split-based evolutionary framework for vehicle routing. In *IERC 2008: Industrial Engineering Research Conference*, Vancouver (Canada), May 17-21, 2008.

– C. A. Amaya, A. L. Medaglia, J. E. Mendoza, N. Velasco, J. G Villegas. Avances en problemas de ruteo. *V V Regional Congress of the Institute of Industrial Engineering (Region 16)*, Paipa (Colombia), March 28-29, 2008.

Appendix B

French summary of the thesis: Problèmes de tournées de véhicules avec remorques

Introduction

Les opérations de transport se retrouvent à différents stades des chaînes d'approvisionnement, de la collecte des matières premières à la distribution des produits finis. Dans les pays européens, le transport représente 19% à 37% du coût logistique total pour plusieurs industries [56]. Alors, il n'est pas surprenant que la recherche operationelle soit appliquée avec succès pour modéliser et optimiser différentes décisions de transport. Parmi elles, les décisions de routage sur les nœuds d'un réseau se rencontrent dans différents environnements tels que la livraison de boissons gazeuses [69,101], la confection des tournées de vendeurs [68], la collecte du lait [24], la livraison des produits laitiers [120], la livraison des produits surgelées [44], la livraison de viandes [15], les opérations de recyclage [13,104], le transport scolaire [29], la livraison du courrier [121], le ramassage des déchets industriels [109], les soins de santé à domicile [43], la livraison de béton prémélangé [120], les opérations de maintenance dans les services publics [81], l'approvisionnement en composants chez les fournisseurs [1], la livraison de produits finis divers [95,125] et les services après-vente [20,125], entre autres. Cependant, tous ces décisions partagent la même structure sous-jacente dans laquelle un ensemble de serveurs mobiles (des véhicules en général) répond à des demandes pour un bien (ou un service), provenant d'un ensemble de clients géographiquement dispersés. Cette structure sous-jacente a été formalisée par le problème de tournées de véhicules (*VRP, vehicle routing problem*).

Formellement, le VRP est défini comme un problème d'optimisation dans lequel des véhicules de capacités limitées et basés à un dépôt principal doivent répondre aux demandes d'un ensemble de clients dispersés géographiquement. L'objectif du VRP est de trouver un ensemble de tournées de longueur totale minimum dans lequel chaque client est visité une seule fois. Toutes les tournées commencent et se terminent au dépôt principal et la demande totale des clients visités dans chaque tournée ne dépasse pas la capacité du véhicule affecté. Depuis l'article de Dantzig et Ramser [35], le VRP a attiré l'attention de nombreuses recherches dans le domaine de l'optimisation. En effet, Eksioglu et al. [41] ont compilé plus de 1000 articles liés au VRP dans leur revue de la littérature. De plus, ces auteurs ont trouvé une croissance significative du nombre de publication au cours des deux dernières décennies.

Selon Golden et al. [61] le VRP est l'une des réussites de la recherche opérationnelle. Une enquête récente sur les logiciels commerciaux présente 22 produits utilisés dans différentes industries [91]. Egalement, une étude des applications réelles estime que la réduction des coûts obtenue avec l'utilisation de techniques d'optimisation varie de 7% à 37%, en fonction des caractéristiques du problème de tournées de véhicules et des critères de mesure pour les économies [116]. Actuellement, la nécessité de réduire les gaz à effets de serre donne de nouvelles raisons d'étudier le VRP et ses extensions [110].

Le VRP classique a été étendu avec différentes contraintes afin de modéliser plusieurs caractéristiques trouvées dans la pratique. Une liste non exhaustive des extensions du VRP comprend : le VRP avec distance limitée (une limite maximale s'applique à la longueur totale ou à la durée totale de chaque tournée) [73,79] ; le VRP avec fenêtres horaires, où chaque client doit être visité dans un intervalle de temps prédéfini (*fenêtres horaire*) [30], le VRP avec *backhauls* où des clients à livrer doivent être visités avant des clients de collecte [59,108], le VRP avec collectes et livraisons dans lequel chaque demande de transport a un lieu de ramassage (*pick-up*) et un lieu de livraison (*delivery*) [37,89,90] ; les VRPs avec flotte hétérogène qui prennent en compte une flotte composée de véhicules avec capacités et coûts différente [6,98] ; le VRP avec contraintes d'incompatibilité dans lequel chaque client ne peut être servi que par un sous-ensemble de véhicules d'une flotte hétérogène [26,27] ; le VRP ouvert dans lequel les véhicules ne sont pas tenus de retourner au dépôt après avoir terminé le service [47,77] ; les VRP périodiques, où les clients doivent être visités plusieurs fois pendant un horizon de planification qui s'étend sur quelques jours [31,49], le VRP avec préemption des tâches, où la contrainte de livraison de chaque client par une seule tournée est éliminée [2,22], le VRP avec compartiments, où chaque client demande plusieurs produits, les véhicules ont plusieurs compartiments et chaque compartiment est dédié à un produit [42], le VRP avec plusieurs dépôts (*MDVRP, multi-depot VRP*), où les véhicules sont basés à plusieurs dépôts [31,103], le problème de tournées de camions et remorques (*TTRP, truck and trailer routing problem*), où une flotte hétérogène composée de camions et de remorques doit servir un ensemble de clients avec des contraintes d'incompatibilité [25], entre autres.

De nouvelles variantes du VRP apparaissent également lorsque le routage est intégré avec d'autres décisions stratégiques, tactiques ou opérationnelles. Par exemple, des décisions de routage coexistent avec le placement d'entrepôts dans le problème de localisation-routage [40,86] (*LRP, location-routing problem*) ; le contrôle des stocks et les décisions de routage doivent être effectuées simultanément dans l'*inventory routing problem* [19], et le chargement des véhicules est intégré lors de la conception de leurs tournées dans le problème de tournées de véhicules avec chargement [50,52]. Récemment, le terme *rich VRP* a été introduit pour désigner des problèmes de tournées de véhicules qui comprennent plusieurs caractéristiques pratiques ignorés souvent dans la recherche académique [65]. Hasle et Kloster [66] ont présenté et classifié les extensions du VRP qui émergent dans les applications pratiques.

Il est clair qu'un problème de tournées de véhicules unique n'existe pas, il s'agit plutôt d'une grande famille de problèmes avec une structure commune. Par conséquent, les bonnes méthodes pour le VRP sont celles qui produisent des résultats de bonne qualité en des temps de calcul courts, mais aussi celles qui sont faciles à coder et à comprendre, avec peu de paramètres, et facilement adaptables à la diversité des contraintes trouvées dans les applications réelles [32]. Traditionnellement, les méthodes pour résoudre le VRP ont été classées en trois groupes : (i) les méthodes exactes, (ii) les heuristiques, et (iii) les métaheuristiques.

Les méthodes exactes sont basées sur divers modèles de programmation mathématique du VRP. Par exemple, le modèle de flux de véhicules à deux indices utilise des variables entières pour indiquer le nombre de fois qu'une arête donnée est traversée dans la solution. Les arêtes entre paires de clients ne prennent que des valeurs binaires tant que les arêtes entre le dépôt et les clients peuvent aussi prendre la valeur de 2 quand un véhicule ne sert qu'un seul client à partir du dépôt. Comme le nombre de contraintes de capacité dans cette formulation est exponentiel, les méthodes qui l'utilisent son habituellement des approches de branchement et coupe (*branch-and-cut*). De plus, plusieurs familles d'inégalités valides ont été développées afin de renforcer sa relaxation linéaire, Naddef et Rinaldi [84] les résument. Les méthodes de résolution du VRP basés sur cette formulation sont l'algorithme de Laporte et al. [74] et les algorithmes plus récents de *branch-and-cut* d'Augerat et al. [4] et de Lysgaard et al. [80]. Des formulations à trois indices [60] dans lesquelles le véhicule qui traverse une arête est spécifié n'ont pas eu la même réussite que la formulation à deux indices [72]. Actuellement, les algorithmes de type *branch-and-cut* basés sur des formulations à deux indices sont les meilleures méthodes pour résoudre des VRP dans lesquels la capacité du véhicule est grande par rapport à la demande des clients. En utilisant cette méthode Augerat et al. [4] et Lysgaard et al. [80] ont résolu un VRP avec 135 clients, qui est le plus grand VRP résolu par une méthode exacte jusqu'à présent [11]. Il existe aussi des formulations basées sur des flots, qui utilisent des variables binaires pour indiquer si une arête est utilisée dans la solution et des variables continues pour représenter la charge du véhicule en chaque arête. Les inégalités valides pour la formulation à deux indices sont également valides pour les formulations à flux de produits, et peuvent être utilisées pour renforcer sa relaxation linéaire [8].

Le VRP peut être aussi modélisé comme un problème de partition d'ensemble dans lequel les colonnes correspondent aux tournées réalisables [12]. Toutefois, utiliser cette formulation directement n'est pas possible en raison de la taille exponentielle de l'ensemble des tournées réalisables. Par conséquent, les approches exactes basées sur cette formulation utilisent des techniques de génération de colonnes. Par exemple, Baldacci et al. [7] ont présenté une méthode exacte basée sur une formulation en termes de partitionnement d'ensemble avec des coupes supplémentaires. Cette approche a résolu des problèmes jusqu'à 121 clients. La flexibilité de cette technique a aussi été illustrée par Baldacci et

Mingozzi [9] et Baldacci et al. [5], avec des extensions pour résoudre différentes variantes du VRP avec flotte hétérogène, fenêtres horaires, collectes et livraisons, plusieurs dépôts, un horizon de plusieurs jours et des contraintes d'incompatibilité. Baldacci et al. [10,11] ont réalisé un état de l'art qui compare les méthodes exactes les plus récentes pour le VRP.

Etant donné que le VRP est NP-difficile [76], les méthodes exactes ne résolvent en temps acceptable que des problèmes avec une centaine de clients au maximum [10]. Par conséquent, des heuristiques et des métaheuristiques sont utilisées dans la plupart des applications pratiques, où plusieurs centaines de clients sont visités quotidiennement [62]. Nous allons présenter une brève description de ces méthodes approchées.

La plupart des premières méthodes pour le VRP étaient des heuristiques simples qui permettent de trouver rapidement des solutions de bonne qualité. Laporte et Semet [75] les classent en trois groupes : (i) les méthodes constructives, (ii) les méthodes à deux phases et (iii) les heuristiques d'amélioration.

Les méthodes constructives fusionnent les tournées existantes en utilisant un critère de réduction de coûts, comme dans l'heuristique de Clarke et Wright [28], ou ajoutent séquentiellement des clients aux tournées en utilisant un coût d'insertion, comme dans l'heuristique de Mole et Jameson [83].

Les heuristiques à deux phases décomposent le VRP en une affectation des clients aux tournées et la définition de la séquence des clients dans les tournées. Les méthodes *cluster-first, route-second* regroupent en premier les clients en tournées (*clusters*) qui peuvent être desservis par un seul véhicule, puis résolvent un problème de voyageur de commerce (*TSP-Travelling salesman problem*) pour chaque tournée. Ils existent différentes variantes de cette approche en fonction de la méthode utilisée dans la phase de regroupement. Par exemple, l'heuristique de Gillet et Miller [57] utilise des procédures géométriques intuitives, tandis que l'heuristique de Fisher et Jaikumar [46] résout un problème d'affectation généralisée dans la phase de regroupement. D'autre part, les méthodes *route-first, cluster-second* [14,99], construisent d'abord un tour géant visitant tous les clients, puis le divise optimalement en tournées réalisables en utilisant une procédure de découpage.

Les heuristiques d'amélioration emploient une recherche locale simple pour explorer le voisinage d'une solution du VRP. Ces méthodes opèrent sur des tournées individuelles ou sur plusieurs tournées en même temps. Dans le premier cas, des heuristiques d'amélioration pour le TSP peuvent être utilisées, par exemple, les classiques Or-opt [88], 2-opt[48] et 3-opt [34]. Le deuxième cas comprend plusieurs procédures d'échange d'arêtes et de nœuds ; Kindervater et Savelsbergh [70] les ont classées en *node relocation, node exchange* et *edge* *crossover*. Plus récemment, Funke et al. [51] ont fait en état de l'art de la plupart des opérateurs de recherche locale pour des problèmes de tournées de véhicules et ont aussi proposé une représentation unifiée qui permet la modélisation de problèmes avec des contraintes complexes. Les heuristiques de recherche locale pour le VRP ont évolué vers les métaheuristiques, qui permettent d'atteindre meilleurs résultats dans les temps de calcul raisonnables. Nous allons maintenant présenter ces méthodes.

Les problèmes de tournées de véhicules font l'objet d'un nombre important d'implémentations réussies de métaheuristiques [54]. Une métaheuristique est une procédure de haut niveau, conçue pour guider autres méthodes heuristiques vers l'obtention des solutions raisonnables aux problèmes difficiles d'optimisation mathématique. Les métaheuristiques tentent d'échapper aux optima locaux (pour les problèmes qui ont plusieurs optima locaux) et/ou de réduire l'espace de recherche [114]. Elles incluent les algorithmes génétiques, le recuit simulé, la recherche avec tabous, la recherche à voisinage variable (*VNS, Variable Neighborhood Search*), la recherche locale itérative (*ILS, Iterated Local Search*), les stratégies évolutives, les procédures gloutonnes randomisées et adaptatives (*GRASP, Greedy randomized adaptive search procedures*), la recherche dispersée, l'optimisation avec colonies de fourmis, entre autres. Actuellement, des métaheuristiques hybrides combinant des éléments et des principes de différentes métaheuristiques plus classiques permettent de trouver de très bonnes solutions pour plusieurs problèmes d'optimisation combinatoire [21]. Une introduction aux métaheuristiques les plus connues et d'autres sujets connexes sont traités par Gendreau et Potvin [53], et par Glover et Kochenberger [58].

Il existe un grand nombre de métaheuristiques pour la solution du VRP classique. Parmi eux, les meilleures sont basées sur les stratégies évolutives [82,97], les algorithmes mémétiques [18,85,96], la programmation à mémoire adaptative [107,119], l'optimisation par colonies de fourmis [102], la recherche avec tabous [33,36,117] et la recherche adaptative à grand voisinage [94]. Gendreau et al. [54] ont réalisé une revue catégorisée des métaheuristiques pour le VRP et plusieurs de ses extensions. Plus récemment, les matheuristiques combinant les métaheuristiques et les méthodes exactes ont émergé comme une alternative prometteuse pour la solution de différentes variantes du VRP, telles que le VRP classique [107], le VRP avec préemption des taches [3] et le problème de localisation-routage [100], entre autres. Doerner et Schmid [38] ont réalisé une revue des matheuristiques pour le VRP et ses extensions.

Néanmoins, la plupart des métaheuristiques les plus réussies pour les problèmes de tournées de véhicules n'ont été conçues que pour la résolution d'une variante particulière et elles ont beaucoup de composants ou de paramètres. Ce phénomène de « surconception » implique une perte de simplicité et de flexibilité [72]. Ce qui entraîne que les logiciels commerciaux de tournées utilisent encore peu des éléments standards trouvés dans les métaheuristiques conçues par les chercheurs académiques, comme les structures de mémoire et les opérateurs de mutation ou de croisement [115]. Par conséquent, il y a un besoin pour des méthodes flexibles et capables de résoudre différentes variantes du VRP sans beaucoup de modifications, même si cela se fait au prix d'une perte raisonnable sur la qualité des solutions [72].

De plus amples détails sur le VRP, sa modélisation, ses méthodes de résolution, ses extensions et ses applications pratiques sont donnés dans un article de synthèse de Laporte [71], une perspective historique de Laporte [72] et les livres de Toth et Vigo [122] et de Golden et al. [61].

Problèmes de tournées de véhicules avec remorques

Dans cette thèse nous nous sommes intéressés aux problèmes de tournées de véhicules avec remorques, dans lesquels la capacité du camion est augmentée grâce à une remorque. L'utilisation de remorques induit des contraintes d'incompatibilité avec certains clients, à cause d'un espace de manœuvre limitée ou d'une accessibilité via des rues étroites. Ces clients ne peuvent être servis que par le camion seul, sans sa remorque.

Les applications réelles de ce type de problèmes apparaissent dans la distribution et les opérations de collecte dans les zones rurales et les grandes villes. Par exemple, dans plusieurs pays de l'Union Européenne la collecte du lait est réalisée par un camion citerne qui remorque une citerne détachable de plus grande capacité [24,67,123]. Dans ce cas, certaines fermes ne sont pas accessibles avec l'attelage complet et donc la citerne doit être temporairement laissée sur une routes principales avant d'y se rendre. Gerdessen [55] a signalé autres deux applications des problèmes de tournées de véhicules avec remorques aux Pays-Bas, la première pour la distribution d'aliments composés pour animaux dans les régions rurales et la seconde pour la distribution de produits laitiers. Semet et Taillard [113] ont décrit un problème de tournées de véhicules avec remorques, fenêtres horaires, contraintes d'incompatibilité et flotte hétérogène lie aux activités de distribution d'une chaîne de magasins d'épicerie en Suisse.

Les problèmes homologues de tournées sur arcs apparaissent dans la conception de tournées pour la distribution du courrier *parc and loop* [78], où le facteur conduit son véhicule depuis le bureau de poste jusqu'a des lieux de stationnement où il charge son sac pour livrer le courrier en marchant dans les rues. Dans ce cas le facteur correspond au camion et son véhicule à la remorque. La collecte des déchets dans les villages et les petites villes a aussi une structure similaire. Par exemple, à Due Carrare (Italie) [92] les rues étroites sont servies par véhicules de collecte de petite taille tandis que la collecte des déchets dans les rues sans restrictions d'accessibilité est effectuée par des camions plus gros, équipés de compacteurs. Comme la déchetterie est loin de la ville, les petits véhicules de collecte rencontrent les compacteurs au milieu de leurs tournées pour y déverser leur contenu, en évitant ainsi de longs trajets à vide.

Malgré l'applicabilité des problèmes de tournées de véhicules avec remorques, ils sont encore rarement étudiés. Par exemple, la revue de la littérature de Gendreau et al. [54] ne rapporte que trois articles avec métaheuristiques pour ce type de problèmes. D'un autre côté, en dehors de la méthode de relaxation lagrangienne de Semet [112] et le *branch-and-price* de Drexl [39], nous n'avons pas trouvé d'autres méthode exactes pour la résolution des problèmes de tournées de véhicules avec remorques. En conséquence, dans cette thèse nous avons étudié deux problèmes de tournées de véhicules avec remorques, le problème de routage d'un seul camion avec remorque et des dépôts satellites (*STTRPSD, single truck and trailer routing problem with satellite depots*), et le problème de tournées de camions et remorques (*TTRP, truck and trailer routing problem*).

Dans le STTRPSD un camion avec une remorque amovible basé à un dépôt principal doit répondre à la demande d'un ensemble de clients accessible seulement avec le camion. Par conséquent, avant de servir les clients, il est nécessaire de détacher la remorque à des places de stationnement appropriées (appelées *trailer points* ou dépôts satellites), où les produits sont transférés entre le camion et la remorque. Une solution du STTRPSD, est composée d'une tournée de premier niveau partant du dépôt principal (réalisée par le camion avec la remorque) et visitant un sous-ensemble de dépôts satellites, et de plusieurs tournées de deuxième niveau (effectués par le camion), qui partent des dépôts satellites visités dans la tournée de premier niveau pour visiter des sous-ensembles de clients avec une demande totale inférieure a la capacité du camion. Le MDVRP peut être considéré comme un cas particulier du STTRPSD dans lequel las distances entre les dépôts sont nulles. Le STTRPSD peut être aussi vu comme une version simplifiée du problème de localisation-routage à deux niveaux (*2E-LRP, two-echelon location-routing problem*) [63,87] avec un seul véhicule dans le premier niveau et sans coûts fixes pour les dépôts.

Dans le TTRP, une flotte hétérogène fixe de camions et de remorques est utilisée pour répondre à la demande d'un ensemble de client. Les clients sont répartis en *clients-camion* et *clients-véhicule*. Les clients-camion ont des contraintes d'incompatibilité, et ne sont accessibles que par le camion seul, sans la remorque. Au contraire, les clients-véhicule n'ont pas de restrictions d'accessibilité, et leur emplacement peut être utilisé pour garer la remorque avant de servir les clients-camion. Dans le TTRP, il y a donc trois types de tournées : des *tournées de camions pures* effectuées par un camion et sans restrictions d'accessibilité ; *tournées de véhicules pures* effectuées par un camion avec une remorque et ne servant que des clients-véhicule, et des *tournées de véhicule avec sous-tours* réalisées par un camion avec remorque et servant des clients-camion et des clients-véhicule. Les tournées de véhicules avec sous-tours ont une structure semblable au STTRPSD, alors le TTRP généralise en quelque sorte le STTRPSD. Le TTRP est également liée au VRP avec flotte hétérogène fixée [6,118] car il y a deux types de véhicules avec des capacités différentes (le camion et le camion avec la remorque). De la même manière, en raison des contraintes d'incompatibilité des clients-camion, le TTRP est lié au VRP avec contraintes d'incompatibilité [26].

Dans cette thèse, trois variantes complexes du VRP (STTRPSD, MDVRP et TTRP) ont été abordées avec succès en utilisant des procédures de type *route-first, cluster-second*. Ces travaux sont complétés par une dernière partie consacrée à l'élaboration d'une bibliothèque de composants logiciels (*framework*) simple et flexible, pour résoudre des problèmes de tournées de véhicules avec des heuristiques et métaheuristiques basées sur le principe *route-first, cluster-second*. Après avoir analysé les caractéristiques communes des procédures de découpage de tournées, il a été possible d'élaborer un *framework* orienté à objets qui gère les principaux cas d'applications. Grâce à ce logiciel, les utilisateurs peuvent traiter des cas réels du VRP, avec en effort réduit de codage pour produire rapidement des maquettes de logiciels de résolution.

Ce document est divisé en trois parties : la première est consacrée aux métaheuristiques et méthodes exactes pour le STTRPSD. La deuxième partie présente des métaheuristiques hybrides et matheuristiques pour résoudre le TTRP, et enfin la troisième partie présente le *framework* pour le développement des heuristiques *route-first, cluster-second*. Le document est composé de cinq articles de recherche auto-contenus et écrits en collaboration avec différents coauteurs qui sont mentionnés au début de chaque article. Etant auto-contenu, chaque article présente sa propre introduction, ses notations, ses conclusions et références. Finalement, le dernier chapitre présente les conclusions et les perspectives de recherche future. L'annexe A résume les publications élaborées lors de la préparation de la thèse. Le reste de ce document en français donne une brève description de la contribution de chaque chapitre suit.

Chapter II : GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots

Ce chapitre présente d'abord le STTRPSD. Étant donné que le STTRPSD n'a jamais été abordé auparavant, il commence par une formulation de programmation linéaire en nombres entiers pour définir clairement le problème. Puisque le STTRPSD généralise le MDVRP et le VRP, le chapitre décrit plusieurs méthodes heuristiques et métaheuristiques pour le résoudre. La première est une heuristique intuitive de type *cluster-first, route-second* dans laquelle les clients sont affectés au dépôt satellite le plus proche et une heuristique de meilleure insertion est utilisée pour construire les tournées de premier et de deuxième niveau. L'heuristique suivante est une méthode *route-first, cluster-second*. Cette heuristique obtient des solutions du STTRPSD par le découpage optimal de tours géants qui ne visitent que les clients. La procédure de programmation dynamique développée pour le découpage optimale des tours géants construit simultanément les tournées de deuxième niveau, sélectionne les dépôts satellite à ouvrir, puis construit la tournée de premier niveau. La troisième heuristique est une descente à voisinage variable (*VND, variable neighborhood descent*) [64] avec cinq voisinages : les trois premières sont destinées à améliorer le routage dans les tournées tandis que les deux autres améliorent la tournée de premier niveau en ajoutant ou supprimant des dépôts satellites.

La procédure *route-first, cluster-second* et le VND sont les composants utilisés pour deux métaheuristiques pour le STTRPSD. La première est un GRASP/VND qui utilise le VND comme procédure d'amélioration [105]. La seconde est une recherche local évolutive à démarrages multiples (*multi-start evolutionary local search*), dans laquelle une recherche locale évolutive [126] est redémarrée à partir de différentes solutions initiales, obtenues en perturbant fortement le tour géant de la meilleure solution actuelle. Cette méthode alterne entre des tours géants et des solutions complètes du STTRPSD. La mutation et la perturbation sont effectuées sur le tour géant, alors que le VND opère sur des solutions du STTRPSD.

S'agissant d'un nouveau problème, il n'y a pas des jeux d'essais publics pour le STTRPSD. Par conséquent, nous avons généré 32 problèmes-tests avec différentes caractéristiques, disponibles à l'adresse `http://hdl.handle.net/1992/1125`. L'évaluation numérique réalisée sur ces jeux d'essais a montré que la recherche locale évolutive surpasse le GRASP/VND en qualité des solutions et en temps de calcul. Nous avons également testé ces méthodes sur le MDVRP et obtenu des résultats performants avec la recherche locale évolutive à démarrages multiples.

Des versions préliminaires des méthodes décrites dans ce chapitre ont été présentées dans deux conférences internationales :

– J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia and N. Velasco. Metaheuristics for a truck and trailer routing problem. In *CORAL 2009 : III Combinatorial Optimization, Routing and Location Meeting*, Elche (Espagne), 10-12 juin 2009.

– J.G. Villegas, A.L. Medaglia, C. Prins, C. Prodhon, and N. Velasco. GRASP/evolutionary local search hybrids for a truck and trailer routing problem. In *MIC 2009 : The VIII Metaheuristics International Conference*, Hamburg (Allemagne), 13-16 juillet 2009.

Ce dernier travail a été détaillé dans un article publié dans *Engineering Applications of Artificial Intelligence*. La référence complète est :

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5) :780–794, 2010.

Chapter III : A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots

Après avoir développé des métaheuristiques efficaces pour le STTRPSD, ce chapitre décrit une méthode exacte pour le résoudre. Tout d'abord, une nouvelle formulation à deux indices, appropriée pour la résolution avec une méthode de coupes, est présentée. Ensuite, plusieurs familles d'inégalités valides ont été introduites pour renforcer sa relaxation linéaire. Certaines d'entre elles sont des adaptations au STTRPSD d'inégalités valides du VRP [80], du LRP [16] et du problème des voyageurs de commerce multi-dépôt (*MDMTSP, multi-depot multiple TSP*) [17], tandis que quelques autres sont spécifiques au STTRPSD. Ce chapitre présente plusieurs procédures exactes et heuristiques pour la séparation d'inégalités violées. Avec cette formulation et en utilisant des procédures de séparation, nous avons développé un algorithme de *branch-and-cut* pour le STTRPSD.

L'évaluation numérique sur les jeux d'essai introduit dans le chapitre précédent a montré que'un *branch-and-cut* partiel (avec uniquement les variables du premier niveau comme nombres entiers) produit déjà de bonnes bornes inférieures. L'algorithme de branch-and-cut complet est capable de résoudre (en moins de 15 minutes) des instances avec un maximum de 50 clients et 10 dépôts satellites. La procédure de branch-and-cut complète a également réussi à résoudre optimalement certains problèmes à 100 clients où les clients forment des groupes (clusters), en un temps de calcul de 3 heures. Fait remarquable, pour tous les problèmes résolus par *branch-and-cut*, l'optimalité des meilleures solutions trouvées avec la recherche locale évolutive a été prouvée.

Actuellement, nous travaillons sur l'amélioration de la structure générale de la méthode, et la recherche de nouvelles inégalités valides afin de résoudre des problèmes plus grands. Les résultats préliminaires de l'algorithme de branch-and-cut ont été présentés aux conférences suivantes :

- J.G. Villegas, J. M. Belenguer, E. Benavent, A. Martínez, C. Prins, and C. Prodhon. A cutting plane approach for the single truck and trailer routing problem with satellite depots. In *EURO 2010 : XXIV European Conference on Operational Research*, Lisbon (Portugal), 11-14 juillet 2010.

- J. M. Belenguer, E. Benavent, A. Martínez, C. Prins, C. Prodhon, and J.G. Villegas. A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots. In *SEIO 2010 : XXXII Congreso Nacional de Estadística e Investigación Operativa*, A Coruña (Espagne), 14-17 septembre 2010.

Comme il s'agit d'une recherche en cours, le travail de ce chapitre n'a pas encore été publié en revue mais cela est prévu.

Chapter IV : A GRASP with evolutionary path relinking for the truck and trailer routing problem

Ce chapitre présente une métaheuristique hybride pour le TTRP, méthode qui combine efficacement les éléments du GRASP [45], de la VNS [64] et du *path relinking* [106]. Contrairement à la plupart des méthodes précédentes pour résoudre le TTRP [23,25,111], qui suivent le principe *cluster-first, route-second*, ce chapitre présente une nouvelle procédure *route-first, cluster-second* pour le problème.

La construction randomisée des solutions initiales de la méthode GRASP/VNS hybride est réalisée avec une l'heuristique *route-first, cluster-second*, et une VNS est utilisé pour la phase d'amélioration. Le VNS comprend quatre voisinages visant à améliorer les tournées et sous-tours, plus un voisinages spécialisé pour le TTRP qui sert à améliorer les tournées qui ont une structure de type STTRPSD. Comme le GRASP/VNS explore des solutions réalisables et infaisables, la procédure VNS sert à la fois pour l'amélioration des solutions réalisables et pour la réparation des solutions infaisables. En plus, une procédure de path relinking a été incluse pour améliorer les résultats de l'hybride GRASP/VNS. Trois différentes alternatives ont été testées pour cette PR : en tant que procédure de post-optimisation, comme procédure d'intensification, et sous forme de *evolutionary path relinking* (EvPR).

L'évaluation numérique sur le jeu d'essai introduit par Chao [25] révèle que la PR contribue significativement à la qualité des solutions. Tous les GRASP/VNS avec PR surpassent les résultats des méthodes précédentes de la littérature, ainsi que ceux d'un simple GRASP/VNS sans PR. L'utilisation du *path relinking* comme procédure de post-optimisation offre un bon compromis entre la qualité des solutions trouvées et le temps de calcul. Nonobstant, les meilleurs résultats ont été trouvés avec l'EvPR mais en des temps de calcul augmentés. De plus, le GRASP/VNS avec PR a amélioré 4 des 21 meilleures solutions connues et il est devenu la meilleure méthode publiée pour le TTRP.

Une version préliminaire de la contribution de ce chapitre a été présentée à TRISTAN VII :

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP/VND with path relinking for the truck and trailer routing problem. In *TRISTAN VII : Seventh Triennial Symposium on Transportation Analysis*, Tromsø (Norvège), 20-25 juin, 2010.

Le travail a ensuite fait l'objet d'un article à paraître dans Computers & Operations Research :

– J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco. GRASP with evolutionary path relinking for the truck and trailer routing problem. Doi : 10.1016/j.cor.2010.11.011, 2010.

Chapter V : A Matheuristic for the truck and trailer routing problem

Motivé par les résultats du chapitre précédent dans lequel le GRASP/VNS émerge comme une bonne alternative pour générer solutions diverses de haute qualité pour les procédures de post-optimisation, ce chapitre présente une méthode maheuristique combinant GRASP/VNS et programmation linéaire en nombres entiers.

Il propose une formulation du TTRP sous forme de problème de partition d'ensemble, utilisée dans une procédure matheuristique à deux phases. Dans une première phase un ensemble de colonnes (tournées) est construit en extrayant les tournées des optima locaux trouvés par le GRASP/VNS, puis une deuxième phase essaie de trouver une meilleure solution en résolvant le problème de partition d'ensemble sur cet ensemble de tournées.

Cette matheuristique obtient de meilleurs résultats que le GRASP/VNS avec path relinking du chapitre précédent, et avec des temps de calcul comparables. Par ailleurs, 7 nouvelles meilleures solutions ont été trouvées par la matheuristique proposée. Néanmoins, il y a encore place pour des améliorations de cette méthode : des méthodes spécialisées pour résoudre le problème de partition d'ensemble et des stratégies de gestion de l'ensemble de tournées sont actuellement en cours d'exploration.

Ce chapitre a été soumis à IESM 2011 (International Conference on Industrial Engineering and Systems Management) qui aura lieu à Metz, (France) en mai 2011.

Chapter VI : A route-first cluster-second computational *framework* for vehicle routing heuristics

Lors du développement des méthodes heuristiques et métaheuristiques des chapitres précédents, nous avons abordé différentes variantes du VRP en utilisant des procédures de type *route-first, cluster-second*. Dans tous les cas, ces méthodes ont obtenu des résultats compétitifs. En plus, il y a un besoin de méthodes simples et flexibles pour résoudre des VRPs avec différentes contraintes sans beaucoup de modifications. Ainsi, ce chapitre présente un *framework* extensible orienté-objet pour le prototypage rapide des méthodes heuristiques. Le *framework* est basé sur le principe *route-first, cluster-second* et fournit à l'utilisateur un ensemble de composants réutilisables qui peuvent être adaptés pour résoudre sa variante spécifique du VRP sans trop d'effort de codage.

La flexibilité du *framework* est illustrée par une stratégie évolutive simple pour résoudre le VRP classique et le TTRP. Bien que cette stratégie évolutive n'est pas destinée à être la meilleure méthode pour le TTRP ou le VRP, elle obtient pour les deux problèmes des résultats aussi bons que ceux de procédures constructives et d'amélioration spécialisées. Le *framework* est accessible au public (`http://copa.uniandes.edu.co/?p=181`) et a été testé sur deux applications réelles avec de bons résultats [93,124].

L'architecture du *framework* et les exemples ont été présentés dans les conférences internationales suivantes :

- J. G. Villegas, N. Velasco, C. Prins, J. E. Mendoza, and A. L. Medaglia. A split-based evolutionary framework for vehicle routing. In *IERC 2008 : Industrial Engineering Research Conference*, Vancouver (Canada), 17-21 mai 2008.

- J. G. Villegas, A. L. Medaglia, J. E. Mendoza, C. Prins, C. Prodhon, and N. Velasco. A split-based framework for the vehicle routing problem. In *CLAIO 2008 : XIV Congreso Latino Ibero Americano de Investigación de Operaciones*, Cartagène (Colombie), 9-12 septembre 2008.

– J. G. Villegas, A. L. Medaglia, C. Prins, C. Prodhon, and N. Velasco. Solving the truck and trailer routing problem with a route-first, cluster-second framework. In *ALIO /INFORMS Joint International Meeting*, Buenos Aires (Argentine), 6-9 juin , 2010.

Références

[1] J. Alegre, M. Laguna, and J. Pacheco. Optimizing the periodic pick-up of raw materials for a manufacturer of auto parts. *European Journal of Operational Research*, 179(3) :736–746, 2007.

[2] C. Archetti and M. G. Speranza. The split delivery vehicle routing problem : a survey. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem : Latest Advances and New Challenges*, pages 103–122. Springer, 2008.

[3] C. Archetti, M. G. Speranza, and M. W. P. Savelsbergh. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, 42(1) :22–31, 2008.

[4] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report RR- 949-M, ARTEMIS-IMAG, Université. Joseph Fourier, Grenoble, 1995.

[5] R. Baldacci, E. Bartolini, A. Mingozzi, and R. Roberti. An exact solution framework for a broad class of vehicle routing problems. *Computational Management Science*, 7(3) :229–268, 2010.

[6] R. Baldacci, M. Battarra, and D. Vigo. Routing a heterogeneous fleet of vehicles. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem : Latest Advances and New Challenges*, pages 3–27. Springer, 2008.

[7] R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2) :351–385, 2007.

[8] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research*, 52(5) :723–738, 2004.

[9] R. Baldacci and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2) :347–380, 2008.

[10] R. Baldacci, P. Toth, and D. Vigo. Recent advances in vehicle routing exact algorithms. *4OR-A Quarterly Journal of Operations Research*, 5 :269–298, 2007.

[11] R. Baldacci, P. Toth, and D. Vigo. Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, 175(1) :213–245, 2009.

[12] M. L. Balinski and R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2) :300 – 304, 1964.

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

[14] J. E. Beasley. Route-first cluster-second methods for vehicle routing. *Omega*, 11 :403–408, 1983.

[15] J. M. Belenguer, E. Benavent, and M. Martinez. Rutarep : a computer package to design dispatching routes in the meat industry. *Journal of Food Engineering*, 70 :435–445, 2005.

[16] J. M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. Wolfler-Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 2010. doi : 10.1016/j.cor.2010.09.019.

[17] E. Benavent and A. Martinez. A branch-and-cut algorithm for the multi depot multiple TSP. In *TRISTAN VII : Seventh Triennial Symposium on Transportation Analysis*, pages 70–73, Tromsø, Norway, June 20-25, 2010.

[18] J. Berger and M. Barkaoui. A new hybrid genetic algorithm for the capacitated vehicle routing problem. *Journal of the Operational Research Society*, 54(12) :1254–1262, 2003.

[19] L. Bertazzi, M. Savelsbergh, and M. G. Speranza. Inventory routing. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem : Latest Advances and New Challenges*, pages 49–72. Springer, 2008.

[20] F. Blakeley, B. Bozkaya, B. Cao, W. Hall, and J. Knolmajer. Optimizing periodic maintenance operations for Schindler Elevator Corporation. *Interfaces*, 33 :67–79, 2003.

[21] C. Blum and A. Roli. Hybrid metaheuristics : an introduction. In C. Blum, M. J. B. Aguilera, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 114 of *Studies in Computational Intelligence*, pages 1–30. 2008.

[22] M. Boudia, C. Prins, and M. Reghioui. An effective memetic algorithm with population management for the split delivery vehicle routing problem. In T. Bartz-Beielstein, M. J. B. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 16–30, 2007.

[23] M. Caramia and F. Guerriero. A heuristic approach for the truck and trailer routing problem. *Journal of the Operational Research Society*, 61 :1168–1180, 2010.

[24] M. Caramia and F. Guerriero. A milk collection problem with incompatibility constraints. *Interfaces*, 40(2) :130–143, 2010.

[25] I.-M. Chao. A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29 :33–51, 2002.

[26] I.-M. Chao, B. Golden, and E. Wasil. A computational study of a new heuristic for the site-dependent vehicle routing problem. *INFOR*, 37 :319–336, 1999.

[27] I.-M. Chao and T.-S. Liou. A new tabu search heuristic for the site-dependent vehicle routing problem. In B. L. Golden, S. Raghavan, and E. A. Wasil, editors, *The Next Wave in Computing, Optimization, and Decision Technologies*, pages 107– 119. Springer, 2005.

[28] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivering points. *Operations Research*, 12 :568–581, 1964.

[29] A. Corberán, E. Fernández, M. Laguna, and R. Martí. Heuristic solutions to the problem of routing school buses with multiple objectives. *Journal of the Operational Research Society*, 53 :427–435, 2002.

[30] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis. VRP with time windows. In P. Toth and D. Vigo, editors, *The vehicle routing problem*, pages 157 – 193. SIAM, 2002.

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

[32] J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53 :512–522, 2002.

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

[34] G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6) :791–812, 1958.

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

[36] U. Derigs and R. Kaiser. Applying the attribute based hill climber heuristic to the vehicle routing problem. *European Journal of Operational Research*, 177(2) :719–732, 2007.

[37] G. Desaulniers, J. Desrosiers, A. Erdmann, M. M. Solomon, and F. Soumis. VRP with pickup and delivery. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 225–242. SIAM, 2002.

[38] K. F. Doerner and V. Schmid. Survey : Matheuristics for rich vehicle routing problems. In M. J. Blesa, C. Blum, G. R. Raidl, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 6373 of *Lecture Notes in Computer Science*, pages 206–221, 2010.

[39] M. Drexl. A branch-and-price algorithm for the truck-and-trailer routing problem. Technical report, Deutsche Post Endowed Chair of Optimization of Distribution Networks - RWTH Aachen University, 2006.

[40] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon. A GRASP X ELS approach for the capacitated location-routing problem. *Computers & Operations Research*, 37(11) :1912–1923, 2010.

[41] B. Eksioglu, A. V. Vural, and A. Reisman. The vehicle routing problem : a taxonomic review. *Computers & Industrial Engineering*, 57(4) :1472–1483, 2009.

[42] A. El-Fallahi, C. Prins, and R. Wolfler-Calvo. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, 35 :1725–1741, 2008.

[43] P. Eveborn, M. Ronnqvist, H. Einarsdottir, M. Eklund, K. Liden, and M. Almroth. Operations research improves quality and efficiency in home care. *Interfaces*, 39(1) :18–34, 2009.

[44] J. Faulin, P. Sarobe, and J. Simal. The DSS LOGDIS optimizes delivery routes for FRILAC's frozen products. *Interfaces*, 35 :202–214, 2005.

[45] T. A. Feo and M. G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2) :109–133, 1995.

[46] M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11 :109–124, 1981.

[47] K. Fleszar, I. H. Osman, and K. S. Hindi. A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, 195(3) :803 – 809, 2009.

[48] M. M. Flood. The traveling-salesman problem. *Operations Research*, 4(1) :61–75, 1956.

[49] P. M. Francis, K. R. Smilowitz, and M. Tzur. The period vehicle routing problem and its extensions. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem : Latest Advances and New Challenges*, pages 73–102. Springer, 2008.

[50] G. Fuellerer, K. F. Doerner, R. F. Hartl, and M. Iori. Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, 201(3) :751 – 759, 2010.

[51] B. Funke, T. Grünert, and S. Irnich. Local search for vehicle routing and scheduling problems : review and conceptual integration. *Journal of Heuristics*, 11(4) :267–306, 2005.

[52] M. Gendreau, M. Iori, G. Laporte, and S. Martello. A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1) :4–18, 2008.

[53] M. Gendreau and J.-Y. Potvin, editors. *Handbook of Metaheuristics*. Springer, 2 edition, 2010. In press.

[54] M. Gendreau, J.-Y. Potvin, O. Bräysy, G. Hasle, and A. Løkketangen. Metaheuristics for the vehicle routing problem and its extensions : a categorized bibliography. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem : Latest Advances and New Challenges*, pages 143–170. Springer, 2008.

[55] J. C. Gerdessen. Vehicle routing problem with trailers. *European Journal of Operational Research*, 93 :135–147, 1996.

[56] G. Ghiani, G. Laporte, and R. Musmanno. *Introduction to Logistics Systems Planning and Control*. Wiley, 2004.

[57] B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2) :340–349, 1974.

[58] F. Glover and G. A. Kochenberger, editors. *Handbook of Metaheuristics*. Kluwer, 1 edition, 2002.

[59] M. Goetschalckx and C. Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42 :39–51, 1989.

[60] B. L. Golden, T. L. Magnanti, and H. Q. Nguyen. Implementing vehicle routing algorithms. *Networks*, 7(2) :113–148, 1977.

[61] B. L. Golden, S. Raghavan, and E. Wasil. *The Vehicle routing problem : latest advances and new Challenges*. Springer, 2008.

[62] B. L. Golden, E. Wasil, J. P. Kelly, and I.-M. Chao. The impact of metaheuristics on solving the vehicle routing problem. In T. Crainic and G. Laporte, editors, *Fleet management and logistics*, pages 33–56. Springer, 1998.

[63] J. Gonzalez-Feliu. The N-echelon location routing problem : concepts and methods for tactical and operational planning. Technical Report halshs-00422492, Laboratoire d 'Economie des Transports, Institut des Sciences de l'Homme, 2009.

[64] P. Hansen and N. Mladenovic. Variable neighbourhood search : principles and applications. *European Journal of Operational Research*, 130 :449–467, 2001.

[65] R. Hartl, G. Hasle, and G. Janssens. Special issue on rich vehicle routing problems. *Central European Journal of Operations Research*, 14 :103–104, 2006.

[66] G. Hasle and O. Kloster. Industrial vehicle routing. In G. Hasle, K.-A. Lie, and E. Quak, editors, *Geometric Modelling, Numerical Simulation, and Optimization*, pages 397–435. Springer, 2007.

[67] A. Hoff and A. Løkketangen. A tabu search approach for milk collection in western Norway using trucks and trailers. In *TRISTAN VI : The Sixth Triennial Symposium on Transportation Analysis*, Phuket, Tailand, June 10-15, 2007.

[68] W. Jang, H. H. Lim, T. J. Crowe, G. Raskin, and T. E. Perkins. The Missouri lottery optimizes its scheduling and routing to improve efficiency and balance. *Interfaces*, 36(4) :302–313, 2006.

[69] G. Kant, M. Jacks, and C. Aantjes. Coca-Cola enterprises optimizes vehicle routes for efficient product delivery. *Interfaces*, 38(1) :40–50, 2008.

[70] G. A. P. Kindervater and M. W. P. Savelsbergh. Vehicle routing : handling edge exchanges. In E. H. L. Aarts and J. K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 337–360. Wiley, 1997.

[71] G. Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics*, 54 :811 – 819, 2007.

[72] G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4) :408–416, 2009.

[73] G. Laporte, M. Desrochers, and Y. Nobert. Two exact algorithms for the distance-constrained vehicle routing problem. *Networks*, 14(1) :161–172, 1984.

[74] G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5) :1050–1073, 1985.

[75] G. Laporte and F. Semet. Classical heuristics for the capacitated VRP. In P. Toth and D. Vigo, editors, *The Vehicle Routing Problem*, pages 109–128. SIAM, 2002.

[76] J. K. Lenstra and A. H. G. Rinnooy-Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11 :221–228, 1981.

[77] A. N. Letchford, J. Lysgaard, and R. W. Eglese. A branch-and-cut algorithm for the capacitated open vehicle routing problem. *Journal of the Operational Research Society*, 58(12) :1642–1651, 2006.

[78] L. Levy and L. Bodin. Scheduling of local delivery carriers for the United States Postal Service. In M. Dror, editor, *Arc Routing : Theory, Solutions and Applications*, pages 419–442. Kluwer, 2000.

[79] C.-L. Li, D. Simchi-Levi, and M. Desrochers. On the distance constrained vehicle routing problem. *Operations Research*, 40(4) :790–799, 1992.

[80] J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2) :423–445, 2004.

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

[82] D. Mester and O. Bräysy. Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & Operations Research*, 34(10) :2964 – 2975, 2007.

[83] R. H. Mole and S. R. Jameson. A sequential route-building algorithm employing a generalised savings criterion. *Operational Research Quarterly*, 27(2) :503–511, 1976.

[84] D. Naddef and G. Rinaldi. Branch-and-cut algorithms for the capacitated VRP. In P. Toth and D. Vigo, editors, *The vehicle routing problem*, pages 53 – 84. SIAM, 2002.

[85] Y. Nagata and O. Bräysy. Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks*, 54(4) :205–215, 2009.

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

[87] V.-P. Nguyen, C. Prins, and C. Prodhon. GRASP with learning process for a two-echelon location routing problem. In *8th International Conference of Modeling and Simulation - MOSIM'10*, Hammamet - Tunisia, May 10-12, 2010.

[88] I. Or. *Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking.* PhD thesis, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL., 1976.

[89] S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems : Part I : transportation between customers and depot. *Journal für Betriebswirtschaft*, 58(1) :21–51, 2008.

[90] S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems : Part II : transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58(2) :81–117, 2008.

[91] J. Partyka and R. Hall. Vehicle routing software survey. OR/MS Today, February 2010. http://www.lionhrtpub.com/orms/surveys/Vehicle_Routing/vrss.html.

[92] A. Pia and C. Filippi. A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, 13(2) :125–141, 2006.

[93] C. Pineda. Optimización al problema de entrega de repuestos a la red de concesionarios Sofasa - Bogotá. Master's thesis, Departamento de Ingeniería Industrial Universidad de los Andes, 2009. (In Spanish).

[94] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research.*, 34 :2403–2435, 2007.

[95] C. Prins. Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case. *Journal of Mathematical Modelling and Algorithms*, 1 :135–150, 2002.

[96] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31 :1985–2002, 2004.

[97] C. Prins. A GRASP X Evolutionary Local Search hybrid for the vehicle routing problem. In F. B. Pereira and J. Tavares, editors, *Bio-inspired Algorithms for the Vehicle Routing Problem*, pages 35–53. Springer, 2009.

[98] C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22(6) :916–928, 2009.

[99] C. Prins, N. Labadi, and M. Reghioui. Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research*, 47(2) :507–535, 2009.

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

[101] J. Privé, J. Renaud, F. F. Boctor, and G. Laporte. Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society*, 57 :1045–1052, 2006.

[102] M. Reimann, K. Doerner, and R. F. Hartl. D-Ants : Savings Based Ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, 31(4) :563–591, 2004.

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

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

[105] M. G. C. Resende. Metaheuristic hybridization with greedy randomized adaptive search procedures. In Z.-L. Chen and S. Raghavan, editors, *TutORials in Operations Research*, pages 295–319. INFORMS, 2008.

[106] M. G. C. Resende, C. C. Ribeiro, F. Glover, and R. Martí. Scatter search and path-relinking : fundamentals, advances, and applications. In M. Gendreau and J. Y. Potvin, editors, *Handbook of Metaheuristics*, pages 87–107. Springer, 2010.

[107] Y. Rochat and E. Taillard. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1(1) :147–167, 1995.

[108] S. Ropke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3) :750–775, 2006.

[109] S. Sahoo, S. Kim, B.-I. Kim, B. Kraas, and A. Popov Jr. Routing optimization for waste management. *Interfaces*, 35(1) :24–36, 2005.

[110] A. Sbihi and R. W. Eglese. Combinatorial optimization and green logistics. *4OR-A Quarterly Journal of Operations Research*, 5(2) :99–116, 2007.

[111] S. Scheuerer. A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33 :894–909, 2006.

[112] F. Semet. A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research*, 61 :45–65, 1995.

[113] F. Semet and E. Taillard. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research*, 41 :469–488, 1993.

[114] E. A. Silver. An overview of heuristic solution methods. *Journal of the Operational Research Society*, 55(9) :936–956, 2004.

[115] K. Sörensen, M. Sevaux, and P. Schittekat. Multiple neighbourhood search in commercial VRP packages : evolving towards self-adaptive methods. In C. Cotta, M. Sevaux, and K. Sörensen, editors, *Adaptive and Multilevel Metaheuristics*, volume 136 of *Studies in Computational Intelligence*, pages 239–253. Springer, 2008.

[116] C. Sutcliffe and J. Board. The ex-ante benefits of solving vehicle-routeing problems. *Journal of the Operational Research Society*, 42(2) :135–143, 1991.

[117] E. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8) :661–673, 1993.

[118] E. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO*, 33(1) :1–14, 1999.

[119] C. D. Tarantilis. Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research*, 32(9) :2309–2327, 2005.

[120] C. D. Tarantilis and C. T. Kiranoudis. A flexible adaptive memory-based algorithm for real-life transportation operations : two case studies from dairy and construction sector. *European Journal of Operational Research*, 179 :806–822, 2007.

[121] C. D. Tarantilis, C. T. Kiranoudis, and N. C. Markatos. Use of the BATA algorithm and MIS to solve the mail carrier problem. *Applied Mathematical Modelling*, 26 :481–500, 2002.

[122] P. Toth and D. Vigo. *The Vehicle Routing Problem.* SIAM, Philadelphia, 2002.

[123] R. Vahrenkamp. Transportation logistic in rural setting. Technical Report 5/1989, Department of Economics, University of Kassel, 1989.

[124] M. A. Vargas, J. L. Walteros, and A. L. Medaglia. Hybrid approach for airline crew pairing optimization for small carriers. In *IERC 2009 : Industrial Engineering Research Conference*, Miami (US), May 30 - June 3, 2009.

[125] D. Weigel and B. Cao. Applying GIS and OR techniques to solve Sears technician-dispatching and home delivery problems. *Interfaces*, 29(1) :112–130, 1999.

[126] S. Wolf and P. Merz. Evolutionary local search for the super-peer selection problem and the p -hub median problem. In T. Bartz-Beielstein, M. J. B. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 1–15, 2007.