Multi-Compartment Vehicle Routing Problems in the Context of Glass Waste Collection

Schriftliche Promotionsleistung
zur Erlangung des akademischen Grades
Doctor rerum politicarum

vorgelegt und angenommen
an der Fakultät für Wirtschaftswissenschaft
der Otto-von-Guericke-Universität Magdeburg

Verfasser:	Tino Henke, M.Sc.
Geburtsdatum und -ort:	27.03.1986 in Magdeburg
Arbeit eingereicht am:	01.12.2017

Gutachter der schriftlichen Promotionsleistung:
Prof. Dr. Gerhard Wäscher
Prof. Dr. Gudrun P. Kiesmüller

Datum der Disputation: 14.02.2018

Acknowledgements

While writing this thesis, I was fortunate to experience a lot of valuable support from many people in my professional and private life. I would like to take the opportunity to acknowledge the impact of those people on the successful submission and defense of this thesis.

First and foremost, my gratitude goes to my academic supervisor, Professor Dr. Gerhard Wäscher, who was not only co-author of most articles included in this thesis but also a highly supportive mentor. During my time at his department at the Otto-von-Guericke-Universität Magdeburg, he counselled me extensively and patiently. Thanks to him, I was able to acquire many professional skills which have been essential for the successful completion of my doctorate and which will surely remain valuable in the years to come. I would also like to thank my former and current colleagues at the department, Professor Dr. Andreas Bortfeldt, Dr. Sebastian Henn, Mahrokh Javadi, Henriette Koch, Dr. Sören Koch, Heike Luka, Dr. André Scholz, and Anke Schwerdtfeger for a friendly, supportive, and co-operative working environment throughout the last years. Moreover, I would like to thank Prof. Dr. Gudrun P. Kiesmüller for reviewing this thesis, as well as the not-yet mentioned members of the defense committee, Prof. Dr. Jan Fabian Ehmke and Prof. Dr. Peter Reichling.

Already from my second Ph.D. year onwards, I was lucky enough to find a second scientific home in Brescia, Italy, where I spent a total of five research visits at the Università degli Studi di Brescia. Professor M. Grazia Speranza and her team at the Department of Economics and Management always hosted me with a lot of friendliness and assistance for which I am deeply grateful. I gained valuable insights into heuristic and exact solutions approaches for routing problems, and along the way I also learned to love the wonderful country of Italy.

Apart from the professional world, I also received support from many people in my private life who provided me with some welcome distraction and who put me on the right track from time to time. I would like to especially thank my partner, Christian Stahlmann, for always encouraging me to push onwards, my parents, Gabriele Henke and Dietmar Montag, for always showing me a lot of optimism, and my dear friends Geeske and Joachim Döring for always spending important quality time with me. I would like to further thank my grandparents, Bärbel Henke and Horst Henke, the remaining members of the Henke and Stahlmann families, as well as all my friends not explicitly mentioned for accompanying me throughout the last years.

Tino Henke

Magdeburg, March 2018

Contents

I

Introduction

Introduction

Vehicle routing problems represent one of the most intensively studied classes of problems in combinatorial optimization. In general, they deal with the delivery of products to customers with a given vehicle fleet, and involve two core decisions, namely the assignment of customers to vehicles, and the determination of sequences according to which the customers are visited. Since Dantzig and Ramser (1959) studied a vehicle routing problem for the first time, numerous variants have been examined. Most vehicle routing problems have been shown to be NP-hard, i.e. no polynomial algorithms are known which solve problems of an arbitrary size to optimality in polynomial time. Therefore, a large part of the corresponding research focusses on the development of heuristic approaches. Solving vehicle routing problems in a systematic manner by application of Operations Research methods has been shown to generate substantial cost savings compared to manual approaches often found in practice. Hasle and Kloster (2007) report cost saving potentials of 5% to 30%. Moreover, solving vehicle routing problems systematically does not only allow for reducing operational costs, but it also offers the opportunity to increase customer satisfaction by considering detailed delivery preferences. Research on vehicle routing problems had started by regarding very simplistic problem variants, and subsequently, many different aspects of real-world problems have been added and studied over the years. Especially, the introduction of powerful computers in the last decades has shifted the research focus to more complex problem variants. A recent overview on vehicle routing research can be found in Toth and Vigo (2014).

One class of vehicle routing problems which have gained most of their research attention since the first consideration by Brown and Graves (1981) only in recent years deals with multi-compartment vehicle routing problems. Such problems generalize the basic capacitated vehicle routing problem by taking into account different product types which must not be mixed during transportation. Moreover, the considered vehicles feature the special characteristic that their capacity can be divided into a certain number of isolated loading areas, i.e. compartments. In each of these compartments, only a single product type can be transported; thus, allowing for simultaneous transportation of multiple product types while ensuring that different product types do not get mixed. Clearly, when customers have demands for more than only one product type, the use of multi-compartment vehicles can generate substantial cost savings. Figure 1 demonstrates this potential on a very simplified example. In this example, one depot (D) and two customers (C1, C2) are considered which require a shipment of one unit of two product types, each. In the figure, the demands for the product types are illustrated by differently colored squares below the customer nodes. For this example, only a single route to be performed is possible, i.e. going from the depot to C1, then to C2, and finally going back to the depot. If vehicles with only a single compartment are used, each customer must be visited twice, and, therefore, the depicted vehicle route must also be performed twice. If, however, vehicles with two compartments are available, it would be possible to visit each customer only once and perform a single route, only. Thus, the travel distance could be reduced by 50%.

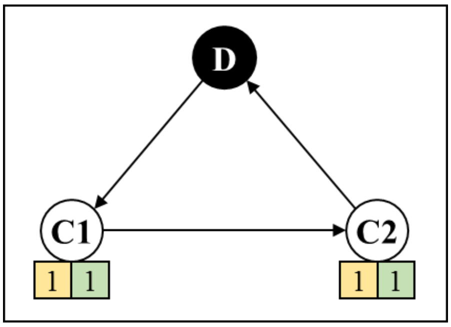


Figure 1: Example of a vehicle routing problem with multiple product types

There are many real-world vehicle routing problems in which different products or product types have to be transported. However, not in all problems it is indeed necessary to consider different product types explicitly in order to determine a feasible solution to the respective planning problem. Often the products to be transported can be reduced to their consumption of the vehicle capacity. This approach is feasible in some contexts; however, there are many problems in which such an aggregated approach is not appropriate. Such situations are found when different product types have varying transportation requirements, or when the product types have a non-solid consistency. The most common example for varying transportation requirements is the occurrence of different temperatures at which product types must be stored and transported, as it is the case in the delivery of grocery products to convenience stores (see Chajakis and Guignard, 2003). In such problems, some product types need to be kept frozen, some product types need to be cooled, whereas other product types require an ambient temperature. With respect to the product consistency, different product types must be considered explicitly to prevent mixing. This is usually the case when liquid or bulk product types have to be transported, e.g. fuel products (see e.g. Brown and Graves, 1981), agricultural products (see e.g. El Fallahi et al., 2008), or waste (see e.g. Muyldermans and Pang, 2010). The latter example also provides the application by which the research in this thesis is motivated, namely the study of multi-compartment vehicle routing problems in the context of glass waste collection.

In many countries, glass waste is disposed in containers which are located at distinct locations in an urban area. At each location, several containers are available into which different types of glass waste can be disposed, e.g. colorless, green, and brown glass. More precisely, each container is dedicated for the disposal of a single glass waste type, only; thus, ensuring that different types of glass waste do not get mixed to support an efficient recycling process. In order to further guarantee the separation of glass waste types during transportation, the corresponding collection vehicles provide a certain number of compartments so that in each compartment a single glass waste type can be transported. Compared to multi-compartment vehicle routing problems studied in the literature, the glass waste collection problem features a property which has rarely been examined so far, namely flexible compartments. In nearly all studied problems, the number and sizes of compartments are fixed for each vehicle, whereas in the glass waste collection problem the number and sizes of compartments are adjustable. Moreover, the type of compartment flexibility may be different. Depending on the characteristics of a vehicle, compartment sizes may be adjusted continuously or in pre-defined discretized steps. Flexible compartments result in additional problem decisions with respect to the number of compartments and their sizes. This thesis investigates the beforementioned variants of compartment flexibility in detail and proposes solution approaches which solve multi-compartment vehicle routing problems with flexible compartment sizes. Furthermore, the glass waste collection problem involves a temporal aspect as it also needs to be decided in which period, e.g. on which day, a specific container should be emptied. Clearly, considering multiple periods increases the complexity of the problem as a further dimension has to be considered. In order to gain some fundamental insights on the glass waste collection problem, single-period variants are studied as well as a multi-period variant. Summarized, the multi-compartment vehicle routing problems considered in this thesis feature two special aspects: Flexible compartments and a multi-period planning horizon. The application of glass waste collection may be considered exemplarily throughout this thesis; however, the analyses are always performed in a general manner and may, therefore, be adapted to other contexts.

The main research objectives of this thesis are (1) to examine different variants of multi-compartment vehicle routing problems, especially with respect to the flexibility of compartments and the length of the considered planning horizon, (2) to propose and study exact as well as heuristic solution approaches which solve these problems, and (3) to gain managerial insights into the impact of using multi-compartment vehicles, different types of compartment flexibility, and the length of the considered planning horizon.

Apart from this introduction (Section 1) and a concluding summary and outlook (Section 7), this thesis consist of five research papers. In Section 2, Henke (2017) introduces a classification scheme for multi-compartment vehicle routing problems, and provides a detailed review on the accumulated research output. Problems previously considered in the literature are very heterogeneous and no comprehensive overview on the research has been available so far. Hence, obtaining an overview on the studied problem variants, distinguishing new research projects from already studied problems, and identifying available benchmark instances has been quite challenging. The introduced classification and review aim at overcoming this lack of overview by providing a structured and comprehensive summary of the many studied variants of multi-compartment vehicle routing problems and by introducing a precise terminology to encourage a more consistent description of multi-compartment vehicle routing problems in future studies.

In Section 3, Henke et al. (2015) study the first out of two single-period variants of the glass waste collection problem. In this variant, compartment sizes are considered to be discretely-flexible, i.e. they can only be varied in discretized steps. This limitation is a specific property of the glass waste collection problem as compartment sizes are adjusted by movable walls which can only be inserted at specific positions within the collection vehicles. This type of compartment flexibility has not been considered before and, therefore, no solution approaches for such problems have been proposed in the literature so far. In this paper, a simple exact solution approach as well as a variable neighborhood search algorithm were developed and tested by means of extensive numerical experiments. In the experiments, the impact of varying problem characteristics, e.g. the number of considered product types or compartments, on the solution quality and the computing times of the algorithms are analyzed. Furthermore, based on some instances adapted from the real-world problem, the benefits of using multi-compartment vehicles instead of single-compartment vehicles are examined.

In Section 4, Henke et al. (2017a) propose an advanced exact solution approach, namely a branch-and-cut algorithm, for the single-period problem with discreetly-flexible compartment sizes. This algorithm was developed to solve large problem instances to optimality and to identify maximal problem specifications for which optimal solutions can be obtained. Numerical experiments on problem instances with up to 50 locations were performed to further study the effect of varying problem characteristics on the solution quality and computing times of the procedure. In addition, the influence of selected components of the algorithm on its performance are analyzed.

In Section 5, Koch et al. (2017) study the second single-period problem variant of the glass waste collection problem in detail. Instead of discretely-flexible compartment sizes, the compartments are considered to be continuously-flexible in this problem, i.e. the sizes can be adjusted to any arbitrary size as long as the vehicle capacity is respected. Clearly, continuous-flexible compartments offer an even more flexible adjustment than discretely-flexible compartments, and might therefore provide the opportunity to further decrease operational costs. Although continuously-flexible compartments are not available in the glass waste collection problem, a detailed analysis of this type of flexibility provides insights into the managerial consequences of acquiring vehicles with discretely-flexible compartments. To solve the problem with continuously-flexible compartments, a simple exact approach was adapted from the discretized problem, and a genetic algorithm was newly developed. The performance of the algorithms was tested extensively. Furthermore, the influence of different degrees of flexibility on operational costs is examined in detail.

In Section 6, the multi-compartment vehicle routing problem with continuously-flexible compartment sizes is eventually considered in a multi-period context by Henke et al. (2017b). Out of all studied problem variants, this problem is closest to the real-world problem of glass waste collection, in which the glass waste containers get filled by a certain amount in each period and decisions about when to empty a container must be made. In the considered single-period variants, this type of decision has been assumed to be given a-priori by a predetermined collection schedule. However, deciding about when to collect glass waste from a container should not necessarily be separated from deciding about vehicle routes. The analyses in this section aim at determining the effect of considering multiple periods simultaneously. In order to do so, a simple exact approach as well as a matheuristic approach were developed to solve the multi-period problem. The performance of both algorithms was tested in numerical experiments. Furthermore, the influence on the operational costs of considering not only one, but multiple periods simultaneously is studied. Finally, the thesis closes with a summary and an outlook on future research possibilities with respect to multi-compartment vehicle routing problems.

References

Brown, G.G.; Graves, G.W. (1981): Real-time dispatch of petroleum tank trucks. In: Management Science 27, 19-32.

Chajakis, E.D.; Guignard, M. (2003): Scheduling deliveries in vehicles with multiple compartments. In: Journal of Global Optimization 26, 43-78.

Dantzig, G.B.; Ramser, J.H. (1959): The truck dispatch problem. In: Management Science 6, 80-91.

El Fallahi, A.; Prins, C.; Wolfler Calvo, R. (2008): A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. In: Computers & Operations Research 35, 1725-1741.

Hasle, G.; Kloster, O. (2007): Industrial vehicle routing. In: Hasle, G. et al. (eds.): Geometric modelling, numerical simulation, and optimization. Berlin, Heidelberg: Springer, 397-435.

Henke, T. (2017): Multi-compartment vehicle routing problems – a review and an extended classification. Working paper, Otto-von-Guericke University Magdeburg.

Henke, T.; Speranza, M.G.; Wäscher, G. (2015): The multi-compartment vehicle routing problem with flexible compartment sizes. In: European Journal of Operational Research 246, 730-743.

Henke, T.; Speranza, M.G.; Wäscher, G. (2017a): A branch-and-cut algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. Working paper, Otto-von-Guericke University Magdeburg. Submitted to Annals of Operations Research.

Henke, T.; Speranza, M.G.; Wäscher, G. (2017b): A matheuristic for the multi-compartment vehicle routing problem with multiple periods. Working paper, Otto-von-Guericke University Magdeburg.

Koch, H.; Henke, T.; Wäscher, G. (2017): A genetic algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. Working paper, Otto-von-Guericke University Magdeburg. Submitted to Journal of Business Economics.

Muyldermans, L.; Pang, G. (2010): On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. In: European Journal of Operational Research 206, 93-103.

Toth, P.; Vigo, D. (2014): Vehicle routing: problems, methods, and applications (2nd ed.). Philadelphia: Society for Industrial and Applied Mathematics.

II

Multi-Compartment Vehicle Routing Problems –
A Review and an Extended Classification

Multi-Compartment Vehicle Routing Problems – A Review and an Extended Classification

Tino Henke

Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany

tino.henke@ovgu.de

Abstract

Among the many extensions of the classical capacitated vehicle routing problem, multi-compartment vehicle routing problems have been studied extensively only in recent years. The available studies, however, often consider substantially different problem variants. As no survey on multi-compartment vehicle routing problems is available so far, the identification of problem similarities and differences between past research studies and potential future projects has been difficult. This paper aims at overcoming this difficulty by proposing an extended classification scheme for multi-compartment vehicle routing problems and extensively reviewing the existing literature. Although, only few identical problems can be identified, some common characteristics among similar applications are observed. Furthermore, managerial findings with respect to the benefits of using multi-compartment vehicles are summarized, and some suggestions for future research directions are proposed.

Keywords: vehicle routing, multiple compartments, review, classification

1 Introduction

Vehicle routing problems have been extensively studied for nearly 60 years. In the standard problem, the capacitated vehicle routing problem (CVRP), the demand of several customers for a single product type must be satisfied by a fleet of homogeneous vehicles with given capacities. In practical situations, however, vehicle routing problems feature more detailed properties, as for example the consideration of time windows (vehicle routing problem with time windows, see e.g. Bräysy and Gendreau, 2005), the limitation of working hours (distance-constrained vehicle routing problems, Li et al., 1992), or the simultaneous occurrence of delivery and collection operations (vehicle routing problems with backhauls, Koç and Laporte, 2018). A comprehensive study about the current research progress on vehicle routing problems is provided by Toth and Vigo (2014). Especially in recent years, even more complex problem variants inspired by real-world applications, so-called rich vehicle routing problems, have gained an increasing research attention (Lahyani et al., 2015b). One class of such rich vehicle routing problems is represented by multi-compartment vehicle routing problems (MCVRPs). MCVRPs occur whenever multiple product types have to be transported which must not be mixed in the same loading area of a vehicle. Typical reasons for separating product types during transportation are either different transportation requirements, e.g. when products need to be transported at different temperatures, or when liquid or bulk product types are regarded. When dealing with such product types, transportation operations can either be performed by using a separate vehicle for each product type or by using vehicles for which the capacity is split or can be split into several loading areas, i.e. compartments. In this way, different product types can be transported simultaneously while ensuring that they are not mixed during transportation. In comparison to the use of vehicles without compartments, substantial operational cost savings may be obtained, especially when customers request several product types simultaneously.

MCVRPs can be identified in many applications, e.g. the collection of different types of waste, the distribution of different types of petroleum products, or the delivery of grocery products to convenience stores. Between most problems considered in the literature so far, the actual problem components may vary substantially, resulting in many studied variants of MCVRPs, whereas only a few papers focus on identical problems. Furthermore, the descriptions of the planning problems differ highly among the studies with respect to the level of detail and terminology used. As no survey on MCVRPs is available so far, obtaining a comprehensive overview of the state of research on MCVRPs has not been simple. However, being aware of already studied problem variants is essential in order to evaluate potential contributions of new research projects. Moreover, when solution approaches are newly developed, the identification and consideration of benchmark instances for similar problem variants support a more reliable evaluation of an algorithm's performance.

In a previous article, a classification of MCVRPs in the context of multi-period fuel distribution problems has been introduced by Coelho and Laporte (2015). However, when planning problems in contexts different from fuel distribution are considered, this classification is not sufficiently detailed as many further problem aspects vary between MCVRPs in the literature. This paper introduces an extended classification and terminology for MCVRPs. Moreover, a comprehensive review on 44 research articles dealing with MCVRPs is presented. By doing so, this paper aims at providing an overview on problem similarities and differences in research studies, encouraging a more consistent description of planning problems in the future, and identifying research gaps. The review focusses on the considered planning problems, the suggested solution approaches, the used data sets, and the obtained managerial insights. Finally, concluding observations with respect to the problem components as well as the managerial insights are presented, and suggestions for further research directions are given.

The paper is structured as follows: In the next subsection, a basic multi-compartment vehicle routing problem is described to provide a basis for the considerations of the remaining paper. In Section 3, a classification of specific problem aspects commonly found in MCVRPs is proposed; thus, allowing for a unified review of problem variants studied in the literature. The actual review on the current state of research with respect to MCVRPs follows in Section 4. For each study, a short summary of the considered planning problem, the proposed solution procedures, the used test instances, and managerial findings is given. In the subsequent Section 5, some observations obtained from the review are discussed, before the paper concludes in Section 6 with suggestions for future research opportunities.

2 A Basic Multi-Compartment Vehicle Routing Problem

Vehicle routing problems are usually either regarded from a distribution perspective, in which goods have to be transported to customers, or from a collection perspective, in which goods are transported from customers. In most cases, however, these two variants do not have an impact on the problem structure and can, therefore, be regarded as identical. In the following, the description of a basic MCVRP is presented with respect to the distribution of products. The described problem represents a basic extension of the CVRP with respect to the core components of MCVRPs, only. In doing so, the common components of MCVRPs are highlighted, and a basis for all remaining considerations in this paper is provided.

In the basic MCVRP, a set of geographically distributed customers is given, each of which has non-negative demands for the delivery of a set of non-mixable product types. Furthermore, one depot with a homogeneous vehicle fleet is given. Each vehicle has an identical capacity that can be separated into compartments, where in each compartment exactly one product type is allowed to be transported. For travelling between each pair of customers or between a customer and the depot, a certain variable transportation cost occurs which may be dependent on the travelling time and/ or the travelling distance between two locations. The amount of each demand for a single product type must be assigned to one or multiple vehicles, and within a vehicle a demand must be assigned to a certain compartment. This consequently results in an assignment of customer locations to vehicles, whereas a location might be visited by multiple vehicles. Furthermore, for each vehicle, a sequence according to which the locations are to be visited has to be determined. In the following, the set of customers visited by a vehicle and the corresponding delivery sequence specify a tour. The objective of the problem is to minimize the sum of all variable transportation costs. Summarized, the basic MCVRP deals with the determination of tours such that all demands are satisfied, all product types remain separated during transportation, no capacity constraints are violated, and the sum of all variable transportation costs is minimized.

Depending on the specific MCVRP, additional aspects, decisions, constraints and components in the objective function may be considered. Especially, the characteristics of demands and compartments often differ between the planning problems studied in the literature. The most common specifications of these problem components will be classified in the following Section, and thus, a unified terminology for MCVRPs will be introduced.

3 Classification Criteria

Recently, Lahyani et al. (2015b) introduced a taxonomy for rich vehicle routing problems. However, with respect to compartments they only distinguish between "compartmentalized" and "not compartmentalized" vehicles. A more detailed classification for MCVRPs in the context of multi-period fuel distribution has been proposed by Coelho and Laporte (2015). In this context, petrol stations have delivery requests for several fuel products, and at each station there is a set of (underground) tanks in which a single fuel type can be stored. For such problems, they distinguish with respect to two criteria: (1) The demand for a single product type may either be supplied by multiple vehicles (split tanks) or it must be supplied by a single vehicle only (unsplit tanks), and (2) the content of a vehicle's compartment may either consist of the delivery quantity for several customers (split compartments), or it may consist of the delivery quantity for one customer only (unsplit compartments). When planning problems in contexts different from fuel distribution are considered, this classification is not sufficiently detailed as further problem aspects vary between MCVRPs in the literature. In the following, additional classification criteria are introduced which relate to commonly found problem characteristics with respect to the delivery requirements and compartment specifications. Apart from the introduced classification criteria, there exist many more differences which are, however, not exclusive to MCVRPs. Thus, for such differences the classification criteria proposed in the taxonomy of Lahyani et al. (2015b) are taken into account.

3.1 Delivery Requirements

When the fulfillment of customer demands is considered, two criteria can be identified which determine the number of times a single customer may be visited. The first criterion defines whether the demands of a single customer for multiple product types must be delivered by one vehicle only, or whether they may be delivered by several vehicles. The second criterion enables an even more precise specification and distinguishes whether a single demand of a customer for a single product type must be delivered by one vehicle only, or whether it may be delivered by several vehicles. In vehicle routing problems with a single product type, only one of these delivery policies must be considered. In the classical CVRP, deliveries for a single demand cannot be performed by several vehicles. However, problems with separate deliveries have also been studied in the literature and are referred to as vehicle routing problems with split deliveries (see e.g. Archetti and Speranza, 2012). When multiple product types are considered, this terminology is not sufficiently precise as it may indicate split deliveries for a single product type or split deliveries for multiple product types. Therefore, a modified terminology for split deliveries in MCVRPs is suggested in the following. When only a single demand for a single product type is regarded, it will be referred to as either unsplit demands or split demands. When demands for multiple product types are regarded, it will be referred to as unsplit or split visits. Consequently, problems with unsplit visits always consider unsplit demands, whereas problems with split visits may either consider split or unsplit demands. The difference between split and unsplit demands has also been identified by Coelho and Laporte (2015) who differ between split and unsplit (fuel) tanks at the customer sites. Summarized, with respect to delivery requirements, the following two classification criteria are introduced:

- D1: Split visits or unsplit visits;
- D2: Split demands or unsplit demands (similar to split and unsplit tanks according to Coelho and Laporte, 2015);

3.2 Compartment Characteristics

With respect to the compartment characteristics, four classification criteria can be identified. First, the number of compartments into which a vehicle's capacity can be divided might either be given in advance (fixed number of compartments) or it might be a decision of the problem (flexible number of compartments). Second, also the sizes of the compartments of a vehicle can be either fixed (fixed compartment sizes), or they might be adjustable for each tour (flexible compartment sizes). Third, the assignment of product types to compartments may be given (fixed assignment of product types to compartments), or it can be a decision of the problem (flexible assignment of product types to compartments). Finally, the content of a compartment may either store the delivery quantities of multiple customers (split compartments), or a compartment may only contain the delivery quantities of a single customer (unsplit compartments). The latter criterion has already been introduced in the classification of Coelho and Laporte (2015), and their terminology is adapted. Summarized, the following four classification criteria with respect to compartment characteristics are considered:

- C1: Fixed or flexible number of compartments;
- C2: Fixed or flexible compartment sizes;
- C3: Fixed or flexible assignment of product types to compartments;
- C4: Split or unsplit compartments (adapted from Coelho and Laporte, 2015).

4 Literature Review

In the following, a comprehensive review on studies about MCVRPs is given. For this, all studies which deal with an MCVRP, which have been written in English language, and which have been published in scientific journals or handbooks have been taken into account. Some studies on vehicles with multiple compartments only focus on the assignment of delivery quantities to vehicles and neglect the determination of delivery sequences within tours, i.e. routes. Apart from some notable exceptions, only papers have been included in this review in which the routing decisions are explicitly considered.

For each of the reviewed articles, the studied MCVRP is classified according to the introduced criteria. As the studied problems may still substantially differ between each other, all problem components which are different from the basic MCVRP described in Section 2 are highlighted, e.g. multiple planning periods, heterogeneous vehicle fleets, or stochastic demands. Furthermore, for each study, the proposed solution approaches, the used test instances, as wells as obtained managerial findings are summarized. A compact overview of all considered research articles is provided in Table 1. In this table, for each study, the corresponding planning problem is classified according to the introduced criteria for MCVRPs, differences to the basic MCVRP are listed, and the components of the objective function are specified. Additionally, for each study, the mentioned practical context as well as the proposed solution procedure are given.

The majority of the reviewed studies deal with quite heterogeneous problems. As problems within the same practical context tend to be slightly less heterogeneous, the review is structured according to applications, namely fuel distribution (Section 4.1), waste collection (Section 4.2), agricultural transportation (Section 4.3), and convenience store delivery (Section 4.4). The remaining studies which consider MCVRPs in a more abstract manner are reviewed in Section 4.5.

4.1 Problems in Fuel Distribution

Brown and Graves (1981) were the first to study vehicles with multiple compartments. However, routing is not explicitly considered in their problem because only a single customer may be assigned to any tour. Instead, the problem focusses on assigning customers to vehicles with respect to many requirements. Each vehicle may perform a sequence of s single-customer trips with respect to a shift length limitation. The considered vehicle fleet is heterogeneous, and the vehicles are located at multiple depots which results in the additional decision of selecting a depot from which a customer should be delivered. The numbers and sizes of compartments in a vehicle are fixed, and the assignment of product types to compartments is flexible. Because the vehicles do not have flow meters by which the amount of fuel to be unloaded can be controlled, the whole content of a compartment has to be unloaded at a single customer, i.e. compartments are unsplit. Furthermore, technical accessibility constraints must be respected, which means that vehicles have different technological equipment and customers have varying technological requirements for unloading products. Hence, not all vehicles are able to deliver all customers. In addition, some product types can only be assigned to specific compartments, and specific loading patterns must be respected when a vehicle's capacity is not fully used, i.e. only the front compartments should be loaded in such cases. The customer demands are deterministic; however, the actual delivery quantities may be adjusted by the distributor to align with the compartment capacities and safety requirements. Each customer may only be visited once, hence, demands and visits are unsplit. Moreover, the customers may have time windows specifying temporal limitations for the deliveries. In the objective function of their problem, the authors consider variable transportation costs as well as penalties if the shift length of a vehicle driver is not fully used or exceeded. To solve the problem, the authors propose a construction and improvement procedure. They further report, that transportation costs in the real-world case could be reduced by about 3% after their system had been implemented. Brown et al. (1987) consider a similar problem in which multiple customers can be visited on a tour. In the objective function, variable transportation costs, balanced workloads, and delivery quantities are taken into account. The authors have developed a computer assisted dispatch system which decomposes the problem into subproblems which are then solved sequentially. Reported results from the case study show that the system lead to annual cost savings for the company of around two million USD.

Another problem with many specific properties is examined by van der Bruggen et al. (1995). They deal with the problem of redesigning the distribution system for a fuel distributor in a multi-period context. Among other aspects, decisions about the opening and closing of intermediate depots, the composition of the vehicle fleet, the determination of delivery patterns to customers, and the planning of delivery routes must be made. The problem is solved by a hierarchical decomposition procedure in which the actual routing problem is solved on the lowest stage. This routing problem represents a multi-period MCVRP in which for each customer, a set of potential delivery patterns with corresponding delivery quantities is known. The vehicle fleet is heterogeneous, and the sizes and numbers of compartments are fixed for each type of vehicle. They consider fixed as well as variable assignments of product types to compartments. Visits and demands seem to be unsplit, whereas compartments seem to be split in this problem. Additionally, time windows, multiple trips per vehicle, shift duration constraints, and the option of overtime are taken into account. The objective of the routing problem is to minimize variable transportation costs, overtime costs, and fixed vehicle costs. The latter component corresponds to a cost which is accounted for whenever a tour is performed, and which is independent of the tour length. After having applied their solution procedure to the overall problem, they determined a solution which suggested that the yearly logistics costs of the distributor could be decreased by 5 to 6%.

Avella et al. (2004) consider a simpler problem with a heterogeneous fleet of vehicles, fixed numbers and sizes of compartments, and a flexible assignment of product types to vehicles. Both, demands and visits, cannot be split. Moreover, compartments are unsplit and may only be filled completely or not at all. Each vehicle may perform several trips while a shift length limitation for each vehicle driver must be respected. The objective in this problem is to minimize the total variable transportation costs. As a solution procedure, the authors propose a simple heuristic approach and an exact branch-and-price algorithm. In their numerical experiments, they use five real-world instances with about 25 customers, and three compartments, each.

A case study in the area of Hong Kong with several problem-specific components is regarded by Ng et al. (2008). After the definition of some assumptions, their problem is reduced to a MCVRP with heterogeneous vehicles, unsplit compartments, fixed compartment numbers and sizes, and a flexible assignment of product types to compartments. Demands in the case study are deterministic, however, as the inventory at the customer sites is managed by the fuel distributor, the actual delivery quantities can be adjusted in a certain interval. Furthermore, demands and visits can be split, and on a single vehicle trip maximally three customers may be visited. In the case study geographical accessibility constraints must be respected, i.e. due to different vehicle sizes not every vehicle is allowed to visit every customer. The objectives consist of maximizing the delivered quantities, as well as minimizing the number of trips and the number of visited customers on a trip. These different objectives are considered in a weighted manner. To solve the case study, the authors developed a decision support system in which a mathematical model for the problem is solved. Based on a single instance they were able to show that their obtained solution improves on the manual solution by reducing the number of trips and increasing the delivered quantities, simultaneously.

Cornillier et al. (2008a) consider a MCVRP with the focus on the so-called tank truck loading problem, in which for a given assignment of customers to vehicles, the assignment of demands to compartments has to be determined. The demands are given in intervals whereat a maximization of delivery quantities is intended. The heterogeneous vehicles have a fixed number of unsplit compartments with fixed compartment sizes, and a flexible assignment of product types to compartments. For the subsequent routing problem, demands and visits are unsplit, and maximally two customers may be visited on any duration-constrained tour. They propose two variants of an exact algorithm which solve the routing aspect of the problem by either complete enumeration or column generation. The algorithm is tested based on 60 instances with up to 200 customers, three product types, and five compartments. In addition, the authors solve a real-world instance with 42 customers and three product types, and they show that the total travelled distance can be decreased by 17.2% compared to the company’s solution. A similar problem is studied by Cornillier et al (2009) in which the maximal number of customers per tour is restricted to four. In addition to Cornillier et al. (2008a), time windows, multiple trips, shift length limitations, and the option of overtime are taken into account. The objective in this problem is to maximize the total profit, which is determined as the difference between all profits generated by the delivered quantities, the variable transportation costs, and the variable overtime costs. To solve the problem, the authors propose two matheuristics in which either the set of available arcs is reduced (arc preselection heuristic), or the set of potential routes is reduced (route preselection heuristic). To evaluate their algorithms, they introduce 40 newly generated instances with up to 50 customers, three product types, and six compartments. Furthermore, they solve one real-world instance with 42 customers and show that they are able to improve the company’s manual result by 22%. Cornillier et al. (2012) extent the beforementioned problem by considering multiple depots, where at each depot a certain fleet of heterogeneous vehicles is available. To solve this problem, they propose an exact method in which all feasible tours are determined in a first step, and a model to select a subset of these tours is used in a second step. The method can also be truncated to a heuristic procedure by restricting the set of preselected tours. They test their algorithms on 60 newly generated instances with 30 customers and up to 6 compartments.

Cornillier et al. (2008b) consider a multi-period MCVRP in which a set of petrol station with underground tanks for storing different types of fuel have to be delivered. The fuel types have deterministic consumptions rates over the planning horizon, and it needs to be ensured that the replenishment is performed in such a way that safety stocks and tank capacities are respected. For the distribution, a heterogeneous fleet of vehicles with fixed numbers and sizes of compartments, and a flexible assignment of product types to compartments is available. Moreover, demands and visits may be split, whereas compartments are unsplit. On each tour maximally two customers may be visited. The problem further considers multiple trips, shift durations, and the option of overtime. In the objective function, revenues generated by the fuel deliveries, variable transportation costs, variable drivers' wages, and variable overtime costs are taken into account. They developed a multi-phase heuristic which solves the occurring subproblems iteratively. For their numerical experiments, they introduce 100 newly generated instances with up to 200 customers, three product types, six compartments, and 28 periods.

Another multi-period problem with deterministic consumption rates and inventories at the customer sites is studied by Popović et al. (2012). The considered vehicle fleet is homogeneous, the number and sizes of compartments is fixed, and the assignment of product types to compartments is flexible. Moreover, compartments are unsplit, and each compartment must always be filled completely or not at all. Each petrol station may only be visited once per period, and the number of stations which can be visited on a single tour is limited by three. In the considered objective function, variable transportation costs as well as inventory holding costs are taken into account. The authors propose a variable neighborhood search algorithm to solve the problem. For their tests, they introduce 30 newly generated instances with up to 20 customers, three product types, five compartments, and five periods. Vidović et al. (2014) analyze a similar problem in which the maximal number of customers per tour is limited to four. Furthermore, the objective function additionally takes fixed vehicle costs into account. The authors introduce a heuristic approach which combines a matheuristic component for the determination of an initial solution and a variable neighborhood descent algorithm to gain further improvements. For their experiments, they introduce 200 new instances with up to 50 customers, three product types, four compartments, and five periods.

Coelho and Laporte (2015) deal with several variants of multi-period MCVRPs, in which petrol stations have deterministic consumption rates of fuel products in each period, and in which tanks for storing the fuel products are available at the petrol stations. Therefore, the objective function does consist of variable transportation costs and inventory holding costs. The authors consider four problem variants in which demands and compartments are either split or unsplit. The available vehicle fleet is heterogeneous, and the numbers and sizes of compartments are fixed. For the assignment of product types to compartments also both alternatives are considered. Furthermore, visits to the petrol stations can be split in all variants. For all resulting problem combinations, they propose a mathematical model and a large number of valid inequalities and symmetry breaking constraints. To solve the problem variants, they propose a branch-and-bound algorithm for small problem sizes and a branch-and-cut algorithm for larger problem sizes. To test their algorithms, they generated three batches of instances: (1) ten single-period, single-product-type instances with up to 50 locations, 4 compartments and 18 vehicles, (2) four multi-period, single-product-type instances with up to 20 locations, 4 compartments, 8 vehicles, and 3 periods, and (3) 15 multi-period, multi-product-type instances with up to 20 locations, 3 product types, 5 compartments, 9 vehicles, and 5 periods. From a managerial perspective, the authors also analyze the impact of split or unsplit demands and compartments on the total costs. The corresponding results show that the most flexible combination with split demands and compartments results in the lowest total cost. When compartments may be split, but demands have to be unsplit, the average costs increase by about 1%. In the remaining two cases, the cost increase amounts to about 2%, whereby the costs for the most inflexible variant are, on average, slightly higher than for the variant with split demands and unsplit compartments.

Benantar et al. (2016) consider a single-period MCVRP with heterogeneous vehicles, a fixed number of unsplit compartments, fixed compartments sizes, and a flexible assignment of product types to compartments. Demands are deterministic and unsplit, however, they may be adjusted downwards up to a certain threshold. In this problem, each location may be visited multiple times. Moreover, they consider time windows at the customer locations and geographical accessibility constraints with respect to the vehicles. The heterogeneous vehicle fleet consists of vehicles owned by the distributor and vehicles which can be rented externally. Therefore, in addition to variable transportation costs they also consider penalty costs for rented vehicles in the objective function. They propose a tabu search which is tested based on 56 adapted instances from Solomon (1987) for the vehicle routing problem with time windows with two product types and two compartments, as well as on 15 real-world instances with up to 89 customers and 20 vehicles. They further examine the impact of adjustable demands on the solution and find that the total distance can be decreased by 7% on average, and the number of vehicles by 10% on average if demands are allowed to be adjusted downwards by up to 4%.

Finally, Urli and Kilby (2017) study both, a single-period and a multi-period MCVRP. Although the delivery quantities are deterministic for each day of the planning horizon, the consideration of a multi-period approach is motivated by the additionally considered decision about the composition of the heterogeneous vehicle fleet. For each vehicle which can be acquired for the fleet, certain fixed costs occurring in each period are taken into account. Only these newly acquired vehicles are then available for distributing fuel products in each period of the considered planning horizon. Furthermore, the number and sizes of the compartments are fixed, the assignment of product types to compartments is flexible, and the compartments are seemingly unsplit, whereas demands and visits may be split. In addition, they consider multiple trips, duration-constrained tours, time windows, and geographical accessibility constraints. The objective in this problem is to minimize the sum of fixed cost for the fleet mix and variable transportation costs. To solve the problem, the authors suggest a large neighborhood search based on a constraint programming model, which is tested on 25 single-period instances and one multi-period instance obtained from a company. Among others, they analyze the benefit of taking a multi-period planning horizon into account compared to a single-period planning horizon. Based on one instance the decrease in total cost is, however, rather small with 0.6%.

4.2 Problems in Waste Collection

Muyldermans and Pang (2010a) consider a MCVRP in the context of domestic waste collection with a homogeneous fleet of vehicles, fixed numbers and sizes of compartments, and a fixed assignment of product types to vehicles. In this problem, the demands are deterministic and unsplit, whereas visits and compartments may be split. The objective is simple and comprises of the minimization of the total travelled distance. To solve this problem, the authors introduce a guided local search algorithm, which is tested on 19 MCVRP instances from El Fallahi et al. (2008) with up to 483 customers, two product types, and two compartments. Furthermore, based on newly generated instances with up to 300 customers, four product types, and four compartments, they analyze for which problem characteristics the use of multi-compartment vehicles is beneficial in comparison to vehicles with only one compartment. The results show that the use of multi-compartment vehicles becomes more beneficial with an increasing number of product types, an increasing vehicle capacity, decreasing demand sizes relative to the vehicle capacity, an increasing number of customers having a demand for all product types, a decreasing customer density in the geographical space, and an increasing centrality of the depot. An arc-oriented version of the previous problem is studied by Muyldermans and Pang (2010b) for the first time. In this problem, the demand for the collection of different waste types occurs on the arcs instead of in the nodes of the network. The characteristics with respect to the vehicle fleet, compartments, demands, and visits is similar to the previous problem. As a solution procedure, the authors have adapted their guided local search from the node-oriented problem. The algorithm is tested based on 24 instances for the capacitated arc routing problem from Eglese (1994) with up to 190 arcs. For further tests, the same instances have been adapted to the multi-compartment problem, resulting in instances with up to 190 arcs, two product types, and two compartments. In addition to the experiments on the performance of their algorithm, the authors also study the benefits of co-collection for the arc-oriented problem by using several newly generated instances with up to 312 required arcs, four product types, and four compartments. For this problem, the observed impacts of the problem characteristics on the benefits of co-collection are similar to the node-oriented problem.

Also, Reed et al. (2014) examine a MCVRP in domestic waste collection. In this problem, demands are deterministic and unsplit, and each location may only be visited once. Although their problem description is not very precise, they seem to consider a homogeneous fleet of vehicles with a fixed number of split compartments, fixed compartment sizes and a fixed assignment of product types to compartments. The objective of the studied problem is to minimize the total travelled distance. In order to solve the problem, Reed et al. (2014) propose an ant colony system algorithm. For the numerical tests, they derive two instances from one CVRP instance of Christofides et al. (1979). The instances consider 50 customers, two product types, and two compartments.

A problem in a slightly different context, namely the collection of different types of glass waste, is studied by Henke et al. (2015). They consider a homogeneous vehicle fleet, flexible assignments of product types to compartments, and split compartments. As a special feature, the number of compartments is flexible up to a certain maximum. Furthermore, compartment sizes are also flexible, however, the sizes can only be varied in discretized steps. Demands are deterministic and unsplit, whereas visits may be split. The objective is to minimize the total variable transportation costs. To solve the problem, the authors propose a variable neighborhood search, which is tested based on 1,350 newly generated small instances and 27 newly generated larger instances with up to 50 customers, nine product types, and nine compartments. Experiments with respect to solving a mathematical model optimally show that the computing times increase with an increasing number of product types and demands, and a decreasing number of maximal compartments. In further experiments on 25 real-world inspired instances with up to 34 customers, three product types, and maximally two compartments, the authors observe that using vehicles with multiple compartments can decrease the total transportation costs by 34.8%, on average, compared to vehicles with only a single compartment.

Oliveira et al. (2015) deal with an MCVRP in the context of recyclable waste collection. The available vehicle fleet is homogeneous, the number and sizes of compartments as well as the assignment of product types to compartments are fixed, and compartments may be split. Furthermore, demands and visits are unsplit. In the objective function they minimize the total travelled distance. For solving the problem, they propose a cluster-first-route-second approach which groups waste containers to be collected according to their collection frequency. The authors apply the algorithm to one real-world instance with two product types, and two compartments. Their result shows that the total transportation costs can be decreased by 5% if vehicles with multiple compartments are used compared to vehicles with only single compartments. However, the result also indicates that the number of collection tours increases and that the collection tours are less balanced.

Elbek and Wøhlk (2016) are the first to study a multi-period MCVRP in the context of waste collection with many application specific aspects. They consider two types of waste, paper and glass waste, which have to be collected from containers in an urban area by a single vehicle which carries two larger containers, i.e. compartments. Hence, the number and sizes of the split compartments, and the assignment of product types to compartments are fixed. As a special feature, the collected wastes need to be transported to an intermediate facility at which the compartment for paper can be emptied, and the compartment for glass waste can be exchanged with an empty compartment if it is completely filled. If not, the current amount of glass waste remains inside the compartment overnight, thus, leading to a reduced capacity in this compartment for the subsequent period. Furthermore, when a certain amount of waste has been accumulated at the intermediate facility, it needs to be transported to distinct recycling facilities by the vehicle. The authors take stochastic and unsplit demands, as well as unsplit visits into account. In the objective function, they consider variable transportation costs and variable service costs for emptying compartments at the intermediate and recycling facilities. As a solution procedure the authors introduce a variable neighborhood search which has been tested based on real-data instances with 211 locations, two product types, two compartments, and 11 periods.

Rabbani et al. (2016) introduce another problem with specific properties in the context of recyclable waste collection. They consider a heterogeneous vehicle fleet with fixed compartment numbers and sizes, and a fixed assignment of product types to split compartments. The vehicle fleet is further distinguished into internal and external vehicles, of which the latter can be rented at a certain cost. Internal vehicles must always return to one of multiple depots, whereas external vehicle do not need to return to a depot. For all vehicles limitations on the tour duration are considered. Furthermore, for each type of waste a distinct disposal facility has to be visited at the end of each tour before returning to a depot. The demands in this problem are deterministic, and visits are unsplit. In the objective function they consider variable transportation costs, costs for renting external vehicles, and service costs for loading and unloading. The problem is solved by genetic algorithms which are tested based on 15 newly generated instances with up to 180 locations and ten product types, and ten compartments.

Only recently, Gajpal et al. (2017) examined a MCVRP in the context of household waste collection, which takes some ecological aspects into account. More precisely, they consider alternative fuel-powered vehicles with a limited fuel tank capacity, thus resulting in distance-constrained tours. The vehicles are homogeneous, and the number and sizes of split compartments as well as the assignment of product types to compartments are fixed. Moreover, deterministic, unsplit demands and unsplit visits are taken into account. The objective of the problem is to minimize the total travelled distance. To solve the problem, they developed an adapted savings algorithm and an ant colony system algorithm. The evaluation of these algorithms is performed based on 28 adapted CVRP instances from Christofides et al. (1979), Christofides and Eilon (1969), and Fisher (1994) with up to 199 customers, two product types, and two compartments. They further analyze the benefits of using multi-compartment vehicles compared to single-compartment vehicles and find that the total distance decreases by 49.26% on average in the latter case.

4.3 Problems in Agricultural Contexts

Ruiz et al. (2004) introduce a MCVRP in the context of the distribution of different types of animal feed. The consider a problem in which the vehicle fleet is heterogeneous, the numbers and sizes of compartments are fixed, the assignment of product types to compartments is flexible, and the compartments are unsplit. Moreover, the demands are deterministic, and visits must be unsplit. On each tour, maximally six customers may be visited. Additionally, the authors take constraints on the maximal distance of a tour as well as geographical accessibility restrictions into account. The objective function consists of a component for the total travelled distance and a component for the vehicle capacity utilization. The latter component is incorporated by determining the transportation cost per transported product unit in relation to a fixed vehicle cost, which naturally decreases with an increasing capacity utilization. They suggest a decomposition and a set partitioning approach to solve their problem. For evaluating their algorithms, they introduce five real-world inspired instances with up to 44 customers. For these instances, they observe that their algorithms can determine solutions in which the operational costs are decreased by 9 to 11% compared to the manual solution obtained by the company.

Another MCVRP in the context of animal feed distribution is studied by El Fallahi et al. (2008). In this problem the vehicle fleet is homogeneous, and the number and sizes of split compartments as well as the assignment of product types to compartments are fixed. For each tour, a duration limitation is not allowed to be exceeded. The delivery demands are deterministic and unsplit, whereas visits may be split. In the objective function, only variable transportation costs are considered. The authors developed a memetic algorithm with path relinking and a tabu search algorithm to solve the problem. Both methods are tested based on 40 instances with up to 484 customers, two product types and two compartments which have been adapted from CVRP instances from Christofides et al. (1979) and Eilon et al. (1971). In further experiments, they slightly modify the algorithms to handle situations in which visits have to be unsplit. They compare the results for this problem variant with the best-known CVRP solutions and obtain average deviations of up to 2.3%.

Oppen and Løkketangen (2008) consider a multi-period MCVRP in the context of livestock collection with many specific problem components. The collection demands are deterministic, however, the actual period in which the collection is performed can be decided within the limitations of a certain planning horizon. Interdependencies between these collections occur at the depot, i.e. the slaughterhouse, which faces inventory and production capacity constraints. Demands are further unsplit, but visits may be split. The vehicle fleet is heterogeneous, the sizes of split compartments and the assignment of animal types to compartments is flexible. Furthermore, all tours are duration-constrained. Interestingly, the number of compartments depends on the animal type to be collected and the sequence in which the collection is performed. This occurs as some compartments are arranged above each other. If smaller animal types are collected, the height of the vehicle is sufficient to store two levels of animals, whereas for larger types of animals only one level can be used. The upper compartment must always be loaded before the lower compartment is loaded, otherwise it cannot be used. Moreover, some compartments may not be loaded if compartments behind them are already loaded as they cannot be accessed any longer. Finally, the animals may be classified as breeding animals or as animals with diseases. The former must always be collected first on a tour, while the latter must always be collected last on a tour. The objective of the problem is to minimize the total travelled distance. To solve their problem, the authors propose a tabu search heuristic which is tested on two real-world instances with up to 184 customers, three animal types, and 8 vehicles. The authors show that the company's manual solution can be improved by 16.4%, on average, when solved with their algorithm. In addition, they investigated the benefits of using vehicles with multiple compartments and conclude that for their considered instances the total distances are 16.9% smaller when multi-compartment vehicles are used compared to single-compartment vehicles.

Caramia and Guerriero (2010) deal with a MCVRP in the context of milk collection from milk farms to a central warehouse. The vehicle fleet consists of heterogeneous vehicles with a fixed number of split compartments, fixed compartments sizes, and a flexible assignment of product types to compartments. Furthermore, the problem incorporates aspects from the truck and trailer routing problem, i.e. some of the vehicles are composed of a truck and a dispatchable trailer, and moreover, some customers are only accessible with a truck while others can be accessed by the complete vehicle. Consequently, a duration-constrained tour for a vehicle may consist of a main tour, in which customers without any accessibility restriction are visited by the complete vehicle, and some subtours, in which the trailer is uncoupled and customers with accessibility restrictions are visited by the truck only. In the problem, the demands are deterministic and unsplit, whereas visits may be split. In the objective function, the total travelled distance and the number of vehicles used are taken into account. In order to solve this problem, the authors suggest a matheuristic which iteratively solves two relaxed optimization problems: (1) The assignment of customers to vehicles, and (2) the sequencing of the vehicles. They apply their method to one real-world instance with 158 farms, four quality types of milk to be collected, 13 vehicles which can be trucks, only, or trucks with trailers, and maximally five compartments. The reported result improves upon the company's solution by using only 11 vehicles instead of 12 vehicles and by reducing the total travelled distance by 14.4%.

Another distance-constrained MCVRP in the context of milk collection is studied by Sethanan and Pitakaso (2016). In this problem, each customer has a deterministic demand for the collection of one distinct type of milk which is not allowed to get mixed with milk from other customers. Hence, demands and visits may be split, but compartments are unsplit. The available heterogeneous vehicle fleet consists of vehicles with fixed compartment sizes and numbers, and a flexible assignment of product types to compartments. Each tour is duration-constrained. Furthermore, after a compartment has been used, it has to be cleaned before it can be filled again. Therefore, in addition to travelling costs the authors also consider cleaning costs. They propose six variants of differential evolution algorithms to solve the problem, which are compared to each other based on 14 newly generated instances with up to 40 customers and three compartments per vehicle.

Finally, Lahyani et al. (2015a) introduce a multi-period MCVRP in the context of collecting olive oil of different quality grades from oil producers. Although, the quantities of oil to be collected in each period are explicitly given in advance, several periods are regarded simultaneously, because compartments must always be cleaned if olive oil of the lowest quality grade has been collected in one period and olive oil of higher quality is planned to be collected in the next period. Such cleaning activities are necessary in order to prevent contamination of olive oils of better qualities. The authors consider a heterogeneous vehicle fleet with fixed compartment numbers and sizes, and a flexible assignment of product types to split compartments. The demands are unsplit, whereas visits and compartments may be split. In the objective function, total transportation costs, cleaning costs, and fixed vehicle costs are taken into account. The proposed branch-and-cut algorithm is tested on 20 real-world inspired instances with up to 19 customers, three product types, two compartments, and six periods. The results show that their solutions improve the total travelled distance by 7% compared to the company's solution.

4.4 Problems in Convenience Store Delivery

Chajakis and Guignard (2003) study two very specific MCVRP variants in the context of convenience store delivery with three types of items: frozen, refrigerated, and dry items. In both of their studied variants, a homogeneous fleet of vehicles with fixed assignments of product types to compartments are given. The number and sizes of compartments, however, might be flexible. In the first variant, the complete loading area of the vehicle corresponds to a compartment for dry products, only. However, there is an option to add boxes, i.e. further compartments, of a given capacity for frozen and refrigerated products into the dry compartment. This variant, therefore, considers a flexible number of compartments, fixed compartment sizes for frozen and refrigerated products, and a flexible size for dry products as the corresponding compartment's size is reduced when boxes are added. In the second variant, the vehicles are equipped with a moveable bulkhead which can separate the dry area of the vehicle from the frozen area. In addition, a box for refrigerated products may be added to the frozen area. This variant, therefore, considers a flexible number of compartments and flexible sizes for all compartments. In both variants, vehicles have a weight constraint in addition to the typical volume capacity constraints. Moreover, demands and visits are unsplit, whereas compartments may be split. In the objective function, apart from variable transportation costs, cooling costs are taken into account which are related to the number of boxes added to a vehicle. For both problem variants, the authors introduce mathematical models and lagrangean relaxations which consider only the assignment component of the problem and neglect the routing component. The models have been tested by means of 17 newly generated instances with 25 customers, three product types, and three compartments. Furthermore, they consider a real-world instance with 44 customers, and a real-world inspired instance with 352 customers.

Also, Derigs et al. (2011) consider two MCVRP variants, one variant in the contexts of fuel distribution and another variant in the context of retail food distribution. Demands are always deterministic and unsplit, whereas visits may be split. Moreover, the vehicle fleet is homogeneous, and the compartments can be split. The difference between the two applications is related to the flexibility of the compartments and the assignment of product types to compartments. Whereas in the fuel distribution context, the numbers and sizes of compartments are fixed, and the assignment of product types to compartments is not, in the retail food distribution context, the numbers and sizes of the compartments are completely flexible, and the assignment of product types to compartments is fixed. These different characteristics align with the requirements mentioned in other studies with similar contexts. As an extension, they introduce possible incompatibilities between different product types as well as between product types and compartments. The objective function considers the total variable transportation costs, only. For a general problem formulation which incorporates both problem variants, they propose a suite of several algorithmic components, e.g. different construction procedures and different metaheuristic mechanisms. Based on 400 newly generated instances with up to 200 customers, three product types, and five compartments they extensively analyze the performance of the numerous variants of their solution procedure. To further evaluate their method, they compare the performance of their best configuration to results for instances from El Fallahi et al. (2008) and Muyldermans and Pang (2010a).

4.5 Abstract Problems

Archetti et al. (2015) study a problem with multiple product types to be delivered and a homogeneous vehicle fleet. However, they do not explicitly consider compartments, i.e. only the vehicles' total capacities are taken into account. Therefore, the studied problem can be interpreted as a MCVRP with split compartments, flexible compartment numbers, continuously flexible compartment sizes, and a flexible assignment of product types to compartments. The demands are deterministic and unsplit, whereas visits may be split. The objective is to minimize total variable transportation costs. To solve the problem, the authors propose a branch-and-price-and-cut algorithm which is tested based on 64 small instances with 15 customers and 80 mid-size instances with up to 80 customers. The number of product types varies between two and three. Moreover, Archetti et al. (2016) present an extensive study about four variants of vehicle routing problems, three of which can also be classified as MCVRPs with split compartments, flexible compartment numbers, continuously flexible compartment sizes, and a flexible assignment of product types to compartments. All variants consider deterministic demands and homogeneous vehicles. In the first variant, demands as well as visits are unsplit. In the second variant, demands and visits may be split. And in the third variant, demands must be unsplit, but visits may be split. Again, the objective function consists of variable transportation costs, only. By means of worst-case analyses, the authors identify bounds on the maximal ratios of the objective function values between different problem variants and show that total transportation costs decrease with an increasing flexibility. They further investigate the problem variants in numerical experiments based on 464 instances with up to 100 customers and three product types. For their analyses, they applied solution approaches from the literature and a newly proposed branch-and-cut algorithm for the variant with unsplit demands and split visits. Their results indicate that the use of co-collection is in general beneficial if visits can be split and when customers are distributed randomly, when the number of product types increases, when the average number of product types demanded by the customers increases, when the sizes of the demands are more diverse, when the vehicle capacity increases, and when the number of customers decreases.

Repoussis et al. (2007) deal with a MCVRP with deterministic, unsplit demands and unsplit visits. The available vehicle fleet is heterogeneous, the sizes and numbers of compartments are fixed, the assignment of product types to compartment is flexible, and compartments may be split. Additionally, they take given time windows for the deliveries to customers and limitations for tour durations into account. The objective function in this problem minimizes the sum of variable transportation costs and fixed vehicle costs. To solve this problem, the authors propose a hybrid heuristic, which combines components from GRASP and variable neighborhood search. The numerical experiments have been performed on single-product type instances from Liu and Shen (1999) for the fleet size and mix routing problem with time windows and on adapted instances with up to three product types.

Another MCVRP with a heterogenous vehicle fleet is considered by Wang et al. (2014). In this problem, demands are deterministic and unsplit, whereas visits are split. For each vehicle the numbers and sizes of split compartments are fixed; however, the assignment of product types to compartments is flexible. Moreover, they introduce general compatibility constraints to define which pair of product types can or cannot be assigned to the same compartment, and which product type can or cannot be assigned to a certain compartment. In the objective function, the authors only include variable transportation costs. To solve this problem, they developed a reactive guided tabu search which is tested based on eight instances adapted from Taillard (1999) for the heterogeneous fleet VRP. The instances consider up to 200 customers, two product types, and two compartments.

Abdulkader et al. (2015) deal with a MCVRP in which deterministic demands and unsplit visits are regarded. The vehicle fleet is homogeneous, and the number and sizes of the split compartments as well as the assignment of product types to compartments are fixed. Moreover, the tours are duration-constrained, and the objective function only takes variable transportation costs into account. The authors formulate a mixed-integer model and propose a hybrid ant colony algorithm where hybridization refers to the use of local search procedures to improve the solution of ants. For their experiments, they introduce 28 new benchmark instances which are derived from Christofides et al. (1979) for the CVRP by splitting the vehicle capacities into two compartments and the customer demands into demands for two product types. The instances vary with respect to the number of customer locations between 50 and 199 locations. Within their experiments, they also investigate the benefits of using vehicles with multiple compartments and find that the total costs decrease by 41.4% on average if multi-compartment vehicles are used instead of single-compartment vehicles.

Kabcome and Mouktonglang (2015) study a MCVRP which takes soft time windows into account. In this problem, demands are deterministic, and visits are unsplit. The vehicles in the heterogeneous fleet have fixed compartment numbers and sizes, and a flexible assignment of product types to split compartments. In the objective function, the authors consider distance-dependent transportation costs, quantity-dependent transportation costs, and penalties for time window violations. They propose three mathematical models which take different degrees of detail of their problem into account. The models are implemented to solve three newly generated test instances with up to 15 customers, two product types, and three compartments.

Kaabi (2016) introduces a selective MCVRP with time windows. Selective refers to the characteristic that customers must not necessarily be visited. Instead, a profit is generated for each visited customer and the objective is, therefore, to maximize the total profit. The vehicle fleet considered in this problem is homogeneous, the number and sizes of compartments as well as the assignment of product types to compartments is fixed. The Demands are deterministic and unsplit, whereas visits and compartments may be split. Tours are further distance-constrained. To solve the problem, the authors propose a genetic algorithm which uses iterated local search for intensification. The algorithm has been tested based on 56 instances adapted from Solomon (1987), in which 100 customers, two product types, and two compartments are considered.

Alinaghian and Shokouhi (2017) study a MCVRP with multiple depots, in which demands are deterministic and unsplit, however, visits to customers may be split. The vehicle fleet consists of homogeneous vehicles with a fixed number of split compartments, fixed compartment sizes and a fixed assignment of product types to compartments. Furthermore, the tours are distance-constrained. In the objective function, the authors take variable transportation costs and fixed vehicle costs into account. As a solution procedure for this problem, they introduce a variable neighborhood search, an adaptive large neighborhood search, and a hybrid adaptive large neighborhood search where the hybridization refers to the incorporation of components from the variable neighborhood search. To evaluate their algorithms, they introduce a new set of small instances with up to 12 customers, three product types, and three compartments as well as a set of larger instances with up to 360 customers, three product types, and three compartments. The latter set of instances was adapted from multi depot CVRP instances from Cordeau (2013).

Silvestrin and Ritt (2017) consider a MCVRP with a homogeneous fleet, a fixed number of split compartments, fixed compartment sizes, and a fixed assignment of product types to compartments. Visits to customers can be split, however, the deterministic demands cannot be split. Furthermore, the problem is tour duration-constrained and considers the minimization of the total travel time in the objective function. In order to solve the problem, the authors developed an iterated tabu search algorithm which combines iterated local search with tabu search. They evaluate the performance of their algorithm based on adapted CVRP instances from Christofides et al. (1979) and Golden et al. (1998) with up to 483 customers, five product types, and five compartments. Furthermore, they analyze the benefit of allowing split visits to customers over unsplit visits. They find, that with their algorithm the differences are on average below 1%.

Also, Mirzaei and Wøhlk (2017) deal with a MCVRP with a homogeneous vehicle fleet, a fixed number of split compartments with fixed sizes, and a fixed assignment of product types to vehicles. They further consider unsplit and deterministic demands as well as both, split and unsplit visits. In the objective function, only variable transportation costs are considered. For their problems, they propose a branch-and-cut algorithm which is tested based on 189 newly generated instances with up to 100 customers, four product types, and 4 compartments, as well as on 85 instances form Muyldermans and Pang (2010a). Furthermore, they also compare optimal solutions for unsplit visits to optimal solution with split visits and find that average savings of 1.7% can be obtained.

Mendoza et. al (2010) were the first to examine a stochastic version of the MCVRP. More, precisely, they consider stochastic demands with underlying distribution functions. Both, the demands and visits must be unsplit. Apart from this, the problem includes homogeneous vehicles, fixed compartment numbers and sizes, a fixed assignment of products types to compartments, and split compartments. Moreover, the tours are distance-constrained. The objective in this problem is to minimize the expected variable transportation costs. As a solution procedure, they propose a memetic algorithm, which is tested on two instance sets with 180 instances, each. The instances consist of up to 200 customers, three product types, and three compartments, and they are adapted from an instance generator for the CVRP with stochastic demands by Mendoza et al. (2008). Furthermore, they experiment on 40 instances which were adapted from El Fallahi et al. (2008). In a second paper, Mendoza et al. (2011) introduce three construction procedures for the same problem variant as well as an improvement procedure. The algorithms are tested on the 360 introduced instances from their earlier paper. Goodson (2015) studies a similar MCVRP with stochastic demands. In contrast to the characteristics of the previous two problems, they take limitations of tour durations instead of distances into account. Accordingly, the objective is to minimize the expected total travel time. The problem is solved by a simulated annealing procedure which is tested on the instances introduced by Mendoza et al. (2010).

Finally, Huang (2015) studies a problem which extends an MCVRP by a facility location component. Multiple potential depots are regarded, and it has to be decided whether they should be opened or not. Furthermore, customers are divided into pickup and delivery customers which have to be serviced from specific pickup or delivery vehicles. The vehicle fleet is, apart from the distinction into pickup and delivery vehicles, homogeneous. Furthermore, the numbers and sizes of compartments, and the assignment of product types to vehicles are fixed. Demands are stochastic and unsplit, visits are unsplit, and compartments may be split. The considered objective function consists of expected variable transportation costs, fixed vehicle costs, depot opening costs, and expected penalties for violations of vehicle and depot capacities. The problem is solved by a decomposition approach in which three subproblems are solved iteratively. The algorithm is tested on adapted location-routing instances from Prins et al. (2004) and Tuzun and Burke (1999) with up to 600 customers, three product types, and three compartments. The authors also investigate the impact of stochastic demands compared to deterministic demands and observe that costs in the stochastic case always exceed costs in the deterministic case. However, as penalty costs are not regarded in the deterministic case, the actual impact does not become entirely clear.

Study	D1: split visits	D1: unsplit visits	D2: split demands	D2: unsplit demands	C1: fixed number	C1: flexible number	C2: fixed sizes	C2: flexible sizes	C3: fixed assignment	C3: flexible assignment	C4: split comp.	C4: unsplit comp.	Additional problem aspects	Objective function components	Application	Solution approach
	MCVRP-specific classification criteria															
Abdulkader et al. (2015)		✓		✓	✓		✓		✓		✓		route duration-constrained	variable transportation costs		hybrid ant colony optimization
Alinaghian and Shokouhi (2017)	✓			✓	✓		✓		✓		✓		multiple depots, distance-constrained	variable transportation costs, fixed vehicle costs		variable neighborhood search, large neighborhood search, hybrid algorithm
Archetti et al. (2015)	✓			✓		✓		✓		✓	✓			variable transportation costs		branch-and-price-and-cut
Archetti et al. (2016)	✓	✓	✓	✓		✓		✓		✓	✓			variable transportation costs		worst-case-analyses, branch-and-cut
Avella et al. (2004)		✓		✓	✓		✓			✓		✓	heterogeneous vehicles, multi-trip, shift duration-constrained, compartments must always be completely filled	variable transportation costs	fuel distribution	construction heuristic, branch-and-price
Benantar et al. (2016)	✓			✓	✓		✓			✓		✓	heterogeneous vehicles, time windows, adjustable demands, accessibility constraints	variable transportation costs, penalties for one type of vehicles	fuel distribution	tabu search
Brown and Graves (1981)		✓		✓	✓		✓			✓		✓	maximally one customer per tour, multiple depots, heterogeneous vehicles, multi-trip, shift duration-constrained, adjustable demands, time windows, accessibility constraints, specific loading constraints	variable transportation costs, penalties for over/undertime	fuel distribution	construction and improvement heuristics
Brown et al. (1987)		✓		✓	✓		✓			✓		✓	multiple depots, heterogeneous vehicles, multi-trip, shift duration-constrained, adjustable demands, time windows, accessibility constraints, specific loading constraints	variable transportation costs, balanced work load, distribution quantity	fuel distribution	decomposition heuristic
Caramia and Guerriero (2010)	✓			✓	✓		✓			✓	✓		heterogeneous vehicles, route duration-constrained, truck and trailer component, accessibility constraints	travelled distance, number of vehicles	milk collection	matheuristic
Chajakis and Guignard (2003)		✓		✓		✓	✓	✓	✓		✓		weight constraints	variable transportation costs, cooling costs	convenience store delivery	models and Lagrangean relaxations
Coelho and Laporte (2015)	✓		✓	✓	✓		✓		✓	✓	✓	✓	multi-period, heterogeneous vehicles, inventory balance	variable transportation costs, inventory holding cost	fuel distribution	branch-and-bound, branch-and-cut

Study	D1: split visits	D1: unsplit visits	D2: split demands	D2: unsplit demands	C1: fixed number	C1: flexible number	C2: fixed sizes	C2: flexible sizes	C3: fixed assignment	C3: flexible assignment	C4: split comp.	C4: unsplit comp.	**Additional problem aspects**	**Objective function components**	**Application**	**Solution approach**
Cornillier et al. (2008a)		✓		✓	✓		✓			✓		✓	maximally two customers per tour, heterogeneous vehicles, route duration-constrained, adjustable demands	variable transportation costs, delivery quantities	fuel distribution	enumeration and column generation
Cornillier et al. (2008b)	✓		✓		✓		✓			✓		✓	maximally two customers per tour, multi-period, heterogeneous vehicles, shift duration-constrained, multi-trip, overtime, inventory balance	revenue, variable transportation costs, wages, overtime costs	fuel distribution	multi-phase heuristic
Cornillier et al. (2009)		✓		✓	✓		✓			✓		✓	maximally four customers per tour, heterogenous vehicles, multi-trip, shift duration-constrained, adjustable demands, overtime, time windows	profit, variable transportation costs, overtime costs	fuel distribution	matheuristics
Cornillier et al. (2012)		✓		✓	✓		✓			✓		✓	multiple depots, heterogeneous vehicles, multi-trip, shift duration-constrained, adjustable demands, overtime, time windows	profit, variable transportation costs, overtime costs	fuel distribution	exact approach and matheuristic
Derigs et al. (2011)	✓			✓	✓	✓	✓	✓	✓	✓	✓		incompatibility constraints	variable transportation costs	fuel distribution, convenience store delivery	local search based metaheursitics
El Fallahi et al. (2008)	✓			✓	✓		✓		✓		✓		route duration-constrained	variable transportation costs	animal feed distribution	memetic algorithm, tabu search
Elbek and Wøhlk (2016)		✓		✓	✓		✓		✓		✓		stochastic demands, multi-period, intermediate facilities, temporal interdependent vehicle capacities	variable transportation costs, service costs	waste collection	variable neighborhood search
Gajpal et al. (2017)		✓		✓	✓		✓		✓		✓		distance-constrained	travelled distance	waste collection	savings algorithm, ant colony system algorithm
Goodson (2015)		✓		✓	✓		✓		✓		✓		stochastic demands, route duration-constrained	expected travel time		simulated annealing
Henke et al. (2015)	✓			✓		✓		✓		✓	✓		discreteness of compartment sizes	variable transportation costs	waste collection	variable neighborhood search
Huang (2015)		✓		✓	✓		✓		✓		✓		stochastic demands, multiple depots, pickup and delivery component, facility location component	facility opening cost, expected variable transportation costs, fixed vehicle cost, penalties for expected capacity violations		decomposition heuristic

Study	D1: split visits	D1: unsplit visits	D2: split demands	D2: unsplit demands	C1: fixed number	C1: flexible number	C2: fixed sizes	C2: flexible sizes	C3: fixed assignment	C3: flexible assignment	C4: split comp.	C4: unsplit comp.	Additional problem aspects	Objective function components	Application	Solution approach
Kaabi (2016)	✓			✓	✓		✓		✓		✓		distance-constrained, time windows, profits associated with customers	profits		hybrid genetic algorithm with iterated local search
Kabcome and Mouktonglang (2015)		✓		✓	✓		✓			✓	✓		heterogeneous vehicles, soft time windows	distance and vehicle-dependent transportation costs, quantity and vehicle-dependent transportation costs, time window penalties		models
Lahyani et al. (2015a)	✓			✓	✓		✓			✓	✓		multi-period, heterogeneous vehicles, compartment cleaning activities	variable transportation costs, fixed vehicle costs, cleaning costs	olive oil collection	branch-and-cut
Mendoza et al. (2010)		✓		✓	✓		✓		✓		✓		stochastic demands, distance-constrained	expected transportation costs		memetic algorithm
Mendoza et al. (2011)		✓		✓	✓		✓		✓		✓		stochastic demands, distance-constrained	expected transportation costs		three constructive heuristics, improvement procedure
Mirzaei and Wøhlk (2017)	✓	✓		✓	✓		✓		✓		✓			variable transportation costs		branch-and price
Muyldermans and Pang (2010a)	✓			✓	✓		✓		✓		✓			travelled distance	waste collection	guided local search
Muyldermans and Pang (2010b)	✓			✓	✓		✓		✓		✓		arc oriented	travelled distance	waste collection	guided local search
Ng et al. (2008)	✓		✓		✓		✓			✓		✓	maximally three customers per tour, heterogeneous vehicles, adjustable demands, accessibility constraints	number of trips, number of stops on a trip, profit for delivery quantities	fuel distribution	model
Oliveira et al. (2015)		✓		✓	✓		✓		✓		✓			travelled distance	waste collection	cluster first-route second heuristic
Oppen and Lokketangen (2008)	✓			✓		✓	✓			✓	✓		multi-period, heterogeneous vehicles, route duration-constraints, sequence and product type-dependent capacities, product type-dependent sequences, inventory balance	travelled distance	livestock collection	tabu search

Study	D1: split visits	D1: unsplit visits	D2: split demands	D2: unsplit demands	C1: fixed number	C1: flexible number	C2: fixed sizes	C2: flexible sizes	C3: fixed assignment	C3: flexible assignment	C4: split comp.	C4: unsplit comp.	Additional problem aspects	Objective function components	Application	Solution approach
Popović et al. (2012)		✓		✓	✓		✓			✓		✓	maximally three customers per tour, multi-period, compartments must always be completely filled, inventory balance	variable transportation costs, inventory holding costs	fuel distribution	variable neighborhood search
Rabbani et al. (2016)		✓		✓	✓		✓		✓		✓		multiple depots, heterogeneous vehicles, route duration-constrained, open tours, intermediate facilities	variable transportation costs, fixed costs for one type of vehicles, service costs	waste collection	hybrid genetic algorithms
Reed et al. (2014)		✓		✓	✓		✓		✓		✓			travelled distance	waste collection	ant colony system
Repoussis et al. (2007)		✓		✓	✓		✓			✓	✓		heterogeneous vehicles, route duration-constrained, time windows	fixed vehicle costs, travelled distance		hybrid GRASP/ variable neighborhood serach algorithm
Ruiz et al. (2004)		✓		✓	✓		✓			✓		✓	maximally six customers per tour, heterogeneous vehicles, distance-constrained, accessibility constraints	travelled distance, capacity utilization	animal feed distribution	decomposition heuristic
Sethanan and Pitakaso (2016)	✓		✓		✓		✓			✓		✓	heterogenous vehicles, route duration-constrained, cleaning activities	variable transportation costs, cleaning costs	milk collection	differential evolution algorithms
Silvestrin and Ritt (2017)	✓			✓	✓		✓		✓		✓		route duration-constrained	travelled time		iterated tabu search
Urli and Kilby (2017)	✓		✓		✓		✓			✓		✓	single period and multi-period, heterogenous vehicles, multi-trip, route duration-constrained, accessibility constraints, time windows	variable transportation costs, fixed vehicle costs	fuel distribution	large neighborhood search
van der Bruggen et al. (1995)		✓		✓	✓		✓		✓	✓	✓		multiple depots, multi-period, heterogeneous vehicles, multi-trip, shift duration-constrained, time windows, overtime	variable transportation costs, fixed vehicle costs, overtime costs	fuel distribution	decomposition heuristic
Vidović et al. (2014)		✓		✓	✓		✓			✓		✓	maximally four stations per tour, multi-period, compartments must always be completely filled, inventory balance	variable transportation costs, inventory holding costs, fixed vehicle costs	fuel distribution	matheuristic
Wang et al. (2014)	✓			✓	✓		✓			✓	✓		heterogenous vehicles, incompatibility constraints	variable transportation costs		reactive guided tabu search

Table 1: Overview of the considered studies on MCVRPs

5 Discussion

The review confirms that MCVRPs discussed in the literature so far are very heterogeneous. Apart from few exceptions, each of the 44 reviewed studies deals with a slightly or extensively different planning problem. In fact, only five problem variants have been studied more than once. Identical problems are considered by El Fallahi et al. (2008) and Silvestrin and Ritt (2017), Muyldermans and Pang (2010a), Derigs et al. (2011) and Mirzaei and Wøhlk (2017), Gajpal et al. (2017) and Abdulkader et al. (2015), Oliveira et al. (2015) and Reed et al. (2014), as well as Mendoza et al. (2010) and Mendoza et al. (2011), respectively.

However, if the problems are clustered according to their applications, some frequent similarities can be identified. With respect to problems in fuel distribution it can be observed, that the numbers and sizes of compartments are always fixed, the assignment of product types to compartments is flexible in nearly all problems, and the content of a compartment can often only be delivered to a single customer. Moreover, demands are often unsplit. In waste collection problems, compartments may always be split, and except for one study, the numbers and sizes of compartments as well as the assignment of waste types to compartments are always fixed. Furthermore, demands are split in all problems. In the studied agricultural problems, the numbers and sizes of compartments are always fixed, whereas the assignment of product types to compartments is flexible in most cases. Compartments and visits may often be split, and demands should usually be collected by a single vehicle, only. An interesting characteristic in grocery distribution different from the other applications is that compartment numbers and sizes may be flexible in both studied problems, however, the assignment of product types to compartments is not. Moreover, compartments are always split, whereas demands are always unsplit.

Also with respect to managerial insights, some valuable findings can be obtained. Several studies compare the results obtained by their solution approaches to previous results from companies. Reported savings range between 3 and 22%, whereas the average cost savings potential over all reported results is around 12%. This quite impressive cost savings potential further demonstrates the value of Operations Research methods to solve real-world vehicle routing problems. Further experiments investigate the benefits of using multi-compartment vehicles. The results suggest that savings of the total considered costs or the total travelling distances between 5 and 49.3%, and an average 29.5% can be obtained if vehicles with multiple compartments are used instead of vehicles with single compartments. This finding clearly justifies the consideration of compartmented vehicles in problems with non-mixable product types. With respect to the flexibility of customer visits, reported results show that savings of 1.7 to 2.2% can be obtained if customers are allowed to be visited multiple times, whereas splitting of demands can generate savings of around 1%, and splitting of compartments can generate savings of around 2% in fuel distribution contexts. Finally, if fuel delivery quantities can be adjusted within certain limitations by the distributor, one study reports savings on the total travelled distance of 7% on average.

6 Future Research Directions

A clear observation obtained from the review is that many problems in fuel delivery or waste collection contexts have been studied. In contrast, only two of the reviewed problems deal with grocery distribution, and the specific applications in an agricultural context have only been investigated twice at most. This finding suggests, that there is a further research potential for MCVRPS in the latter two contexts. This proposition is further strengthened by the observation that some problem specifications, e.g. flexible compartment numbers and sizes, are almost exclusively considered in grocery distribution. Moreover, although the general benefits of using multi-compartment vehicles as well as the benefits of split visits have been analyzed by several authors, a more detailed investigation into the impact of compartment flexibility has not been analyzed, yet. This aspect seems to be another valuable opportunity for further investigation.

A further observation obtained from the review shows that out of all studies, only five papers have considered stochastic demands. In fuel delivery, grocery distribution, and agricultural contexts, demands can justifiable be assumed to be deterministic because requests for deliveries or collections are usually placed by customers with perfect knowledge about the corresponding needs. In waste collection, however, the actual amount to be collected from a container is usually not known in advance. Therefore, a more extensive investigation of MCVRPs with stochastic demands, especially in the context of waste collection, provides another research direction for the future.

Finally, for all identified contexts it can be assumed that deliveries or collections are usually not only requested a single time, but on a more regular basis, e.g. waste containers must usually be emptied every week, and grocery stores must usually be supplied every other day. Therefore, a more extensive analysis into the managerial benefits of considering multi-period planning horizons provides another interesting research opportunity for the future.

References

Abdulkader, M.M.S.; Gajpal, Y.; ElMekkawy, T.Y. (2015): Hybridized ant colony algorithm for the multi compartment vehicle routing problem. In: Applied Soft Computing 37, 196-203.

Alinaghian, M.; Shokouhi, N. (2017): Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. In: Omega (in press), 1-15.

Archetti, C.; Bianchessi, N.; Speranza, M.G. (2015): A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem. In: Computers & Operations Research 64, 1-10.

Archetti, C.; Campbell, A.M.; Speranza, M.G. (2016): Multicommodity vs. single-commodity routing. In: Transportation Science 50, 461-472.

Archetti, C.; Speranza, M.G. (2012): Vehicle routing problems with split deliveries. In: International Transactions in Operational Research 19, 3-22.

Avella, P.; Boccia, M.; Sforza, A. (2004): Solving a fuel delivery problem by heuristic and exact approaches. In: European Journal of Operational Research 152, 170-179.

Benantar, A.; Ouafi, R.; Boukachour, J. (2016): A petrol station replenishment problem: new variant and formulation. In: Logistics Research 9, 1-18.

Bräysy, O.; Gendreau, M. (2005): Vehicle routing problem with time windows, part i: route construction and local search algorithms. In: Transportation Science 39, 104-118.

Brown, G.G.; Graves, G.W. (1981): Real-time dispatch of petroleum tank trucks. In: Management Science 27, 19-32.

Brown, G.G.; Ellis, C.J.; Graves, G.W.; Ronen, D. (1987): Real-time, wide area dispatch of mobil tank trucks. In: Interfaces 17, 107-120.

Caramia, M.; Guerriero, F. (2010): A milk collection problem with incompatibility constraints. In: Interfaces 40, 130-143.

Chajakis, E.D.; Guignard, M. (2003): Scheduling deliveries in vehicles with multiple compartments. In: Journal of Global Optimization 26, 43-78.

Christofides, N.; Eilon, S. (1969): An algorithm for the vehicle-dispatching problem. In: Journal of the Operational Research Society 20, 309-318.

Christofides, N.; Mingozzi, A.; Toth, P. (1979): The vehicle routing problem. In: Christofides, N. et al. (eds.): Combinatorial Optimization. Chichester: Wiley, 315-338.

Coelho, L.C.; Laporte, G. (2015): Classification, models and exact algorithms for multi-compartment delivery problems. In: European Journal of Operational Research 242, 854-864.

Cordeau, J.-F. (2013): Multiple Depot VRP Instances. http://neo.lcc.uma.es/vrp/vrp-%20instances/multiple-%20depot-%20vrp-%20instances/

Cornillier, F.; Boctor, F.F.; Laporte, G.; Renaud, J. (2008a): An exact algorithm for the petrol station replenishment problem. In: Journal of the Operational Research Society 59, 607-615.

Cornillier, F.; Boctor, F.F.; Laporte, G.; Renaud, J. (2008b): A heuristic for the multi-period petrol station replenishment problem. In: European Journal of Operational Research 191, 295-305.

Cornillier, F.; Boctor, F.; Renaud, J. (2012): Heuristics for the multi-depot petrol station replenishment problem with time windows. In: European Journal of Operational Research 220, 361.

Cornillier, F.; Laporte, G.; Boctor, F.F., Renaud, J. (2009): The petrol station replenishment problem with time windows. In: Computers & Operations Research 36, 919-935.

Derigs, U.; Gottlieb, J.; Kalkoff, J.; Piesche, M.; Rothlauf, F.; Vogel, U. (2011): Vehicle routing with compartments: applications, modelling and heuristics. In: OR Spectrum 33, 885-914.

Eglese, R.W. (1994): Routing winter gritting vehicles. In: Discrete Applied Mathematics 48, 231-244.

Eilon, S.; Watson-Gandy, C.D.T.; Christofides, N. (1971): Distribution management: mathematical modelling and practical analysis. London, Griffin.

El Fallahi, A.; Prins, C.; Wolfler Calvo, R. (2008): A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. In: Computers & Operations Research 35, 1725-1741.

Elbek, M.; Wøhlk, S. (2016): A variable neighborhood search for the multi-period collection of recyclable materials. In: European Journal of Operational Research 249, 540-550.

Fisher, M.L. (1994): Optimal solution of vehicle routing problems using minimum K-trees. In: Operations Research 42, 626-642.

Gajpal, Y.; Abdulkader, M.M.S.; Zhang, S.; Appadoo, S.S. (2017): Optimizing garbage collection vehicle routing problem with alternative fuel-powered vehicles. In: Optimization 66, 1851-1862.

Golden, B.L.; Wasil, E.A.; Kelly, J.P.; Chao I-M. (1998): The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In: Crainic, T.G.; Laporte, G. (eds.): Fleet Management and Logistics. Boston: Springer, 33-56.

Goodson, J.C. (2015): A priori policy evaluation and cyclic-order-based simulated annealing for the multi-compartment vehicle routing problem with stochastic demands. In: European Journal of Operational Research 241, 361-369.

Henke, T.; Speranza, M.G.; Wäscher, G. (2015): The multi-compartment vehicle routing problem with flexible compartment sizes. In: European Journal of Operational Research 246, 730-743.

Huang, S.-H. (2015): Solving the multi-compartment capacitated location routing problem with pickup–delivery routes and stochastic demands. In: Computers & Industrial Engineering 87, 104-113.

Kaabi, H. (2016): Hybrid metaheuristic to solve the selective multi-compartment vehicle routing problem with time windows. In: Abraham, A. et al. (eds.): Proceedings of the second International Afro-European Conference for Industrial Advancement AECIA 2015, Advances in Intelligent Systems and Computing 427. Switzerland: Springer, 185-194.

Kabcome, P.; Mouktonglang, T. (2015): Vehicle routing problem for multiple product types, compartments, and trips with soft time windows. In: International Journal of Mathematics and Mathematical Sciences 2015, 1-9.

Koç, Ç.; Laporte, G. (2018): Vehicle routing with backhauls: review and research perspectives. In: Computers & Operations Research 91, 79-91.

Lahyani, R.; Coelho, L.C.; Khemakhem, M.; Laporte, G.; Semet, F. (2015a): A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. In: Omega 51, 1-10.

Lahyani, R.; Khemakhem, M.; Semet, F. (2015b): Rich vehicle routing problems: from a taxonomy to a definition. In: European Journal of Operational Research 241, 1-14.

Li, C.-L.; Simchi-Levi, D.; Desrochers, M. (1992): On the distance constrained vehicle routing problem. In: Operations Research 40, 790-799.

Liu, F.-H.; Shen, S.-Y. (1999): The fleet size and mix routing problem with time windows. In: Journal of the Operational Research Society 50, 721-732.

Mendoza, J.E.; Castanier, B.; Guéret, C.; Medaglia, A.L.; Velasco, N. (2008): An instance generator for vehicle routing problems with stochastic demands. Technical Report 08/1/AUTO, Institut de Recherche en Communications et Cybernétiquede Nantes (IRCCyN). École des Mines de Nantes, France.

Mendoza, J.E.; Castanier, B.; Guéret, C.; Medaglia, A.L.; Velasco, N. (2010): A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. In: Computers & Operations Research 37, 1886-1898.

Mendoza, J.E.; Castanier, B.; Guéret, C.; Medaglia, A.L.; Velasco, N. (2011): Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. In: Transportation Science 45, 346-363.

Mirzaei, S.; Wøhlk, S. (2017): Erratum to: a branch-and-price algorithm for two multi-compartment vehicle routing problems. In: EURO Journal on Transportation and Logistics 6, 185-2018.

Muyldermans, L.; Pang, G. (2010a): On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. In: European Journal of Operational Research 206, 93-103.

Muyldermans, L.; Pang, G. (2010b): A guided local search procedure for the multi-compartment capacitated arc routing problem. In: Computers & Operations Research 37, 1662-1673.

Ng, W.L.; Leung, S.C.H.; Lam, J.K.P.; Pan, S.W. (2008): Petrol delivery tanker assignment and routing: a case study in Hong Kong. In: Journal of the Operational Research Society 59, 1191-1200.

Oliveira, A.D.; Ramos, T.R.P.; Martins, A.L. (2015): Planning collection routes with multi-compartment vehicles. In: Barbosa Póvoa, A.P.F.D.; de Miranda, J.L. (eds.): Operations Research and Big Data, Studies in Big Data 15. Switzerland: Springer, 131-138.

Oppen, J.; Løkketangen, A. (2008): A tabu search approach for the livestock collection problem. In: Computers & Operations Research 35, 3213-3229.

Popović, D.; Vidović, M.; Radivojević, G. (2012): Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. In: Expert Systems with Applications 39, 13390-13398.

Prins, C.; Prodhon, C.; Wolfer Calvo, R. (2004): Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité. Proceedings of the MOSIM' 04, Volume 2. Ecole des Mines de Nantes, France: Lavoisier, 1115-1122.

Rabbani, M.; Farrokhi-asl, H.; Rafiei, H. (2016): A hybrid genetic algorithm for waste collection problem by heterogeneous fleet of vehicles with multiple separated compartments. In: Journal of Intelligent & Fuzzy Systems 2016, 1817-1830.

Reed, M.; Yiannakou, A.; Evering, R. (2014): An ant colony algorithm for the multi-compartment vehicle routing problem. In: Applied Soft Computing 15, 169-176.

Repoussis, P.P.; Tarantilis, C.D.; Ioannou, G. (2007): A hybrid metaheuristic for a real life vehicle routing problem. In: Boyanov, T. et al. (eds.): Numerical Methods and Applications, Lecture Notes in Computer Science 4310. Berlin, Heidelberg: Springer, 247-254.

Ruiz, R.; Maroto, C.; Alcaraz, J. (2004): A decision support system for a real vehicle routing problem. In: European Journal of Operational Research 153, 593-606.

Sethanan, K.; Pitakaso, R. (2016): Differential evolution algorithms for scheduling raw milk transportation. In: Computers and Electronics in Agriculture 121, 245-259.

Silvestrin, P.V.; Ritt, M. (2017): An iterated tabu search for the multi-compartment vehicle routing problem. In: Computers & Operations Research 81, 192-202.

Solomon, M.M. (1987): Algorithms for the vehicle routing and scheduling problems with time window constraints. In: Operations Research 35, 254-265.

Taillard, E.D. (1999): A heuristic column generation method for the heterogeneous fleet VRP. In: RAIRO Operations Research 33, 1-14.

Toth, P.; Vigo, D. (2014): Vehicle routing: problems, methods, and applications (2nd ed.). Philadelphia: Society for Industrial and Applied Mathematics.

Tuzun, D.; Burke, L.I. (1999): A two-phase tabu search approach to the location routing problem. In: European Journal of Operational Research 116, 87-99.

Urli, T.; Kilby, P. (2017): Constraint-based fleet design optimisation for multi-compartment split-delivery rich vehicle routing. In: Becks, J.C. (ed.): Principles and Practice of Constraint Programming, Lecture Notes in Computer Science 10416. Cham et al.: Springer, 414-430.

van der Bruggen, L.; Gruson, R.; Salomon, M. (1995): Reconsidering the distribution structure of gasoline products for a large oil company. In: European Journal of Operational Research 81, 460-473.

Vidović, M.; Popović, D.; Ratković, B. (2014): Mixed integer and heuristics model for the inventory routing problem in fuel delivery. In: International Journal of Production Economics 147, 593-604.

Wang, Q.; Ji, Q.; Chiu, C.-H. (2014): Optimal routing for heterogeneous fixed fleets of multicompartment vehicles. In: Mathematical Problems in Engineering 2014, 1-11.

III

The Multi-Compartment Vehicle Routing Problem with Flexible Compartment Sizes

European Journal of Operational Research 246 (2015) 730–743

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

The multi-compartment vehicle routing problem with flexible compartment sizes

Tino Henke[a,*], M. Grazia Speranza[b], Gerhard Wäscher[a]

[a] Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany
[b] Department of Quantitative Methods, University of Brescia, 25122 Brescia, Italy

ARTICLE INFO

Article history:
Received 8 July 2014
Accepted 6 May 2015
Available online 13 May 2015

Keywords:
Vehicle routing
Multiple compartments
Glass waste collection
Variable neighborhood search
Heuristics

ABSTRACT

In this paper, a capacitated vehicle routing problem is discussed which occurs in the context of glass waste collection. Supplies of several different product types (glass of different colors) are available at customer locations. The supplies have to be picked up at their locations and moved to a central depot at minimum cost. Different product types may be transported on the same vehicle, however, while being transported they must not be mixed. Technically this is enabled by a specific device, which allows for separating the capacity of each vehicle individually into a limited number of compartments where each compartment can accommodate one or several supplies of the same product type. For this problem, a model formulation and a variable neighborhood search algorithm for its solution are presented. The performance of the proposed heuristic is evaluated by means of extensive numerical experiments. Furthermore, the economic benefits of introducing compartments on the vehicles are investigated.

1. Introduction

The vehicle routing problem, which will be discussed in this paper, is a variant of the classic capacitated vehicle routing problem (CVRP; for surveys see Golden, Raghavan, & Wasil, 2008; Laporte, 2009; or Toth & Vigo, 2014) and occurs in the context of glass waste collection in Germany. Glass waste has to be recycled by law and is used as a raw material for the production of new glass products. It has to be taken to recycling stations by the consumers where it is disposed into different containers according to the color of the waste (usually colorless, green and brown glass). Colors are kept separated because the production of new glass products is less cost-intensive if the glass waste is not too inhomogeneous with respect to its color. Trucks, which are located at a depot of a recycling company, pick up the glass waste from the recycling stations. Since they possess a relatively large loading capacity, they can call at several recycling stations before they have to return to the depot. Recent truck models are equipped with a special device which allows for introducing bulkheads in predefined positions of the loading space such that it can be split into different compartments and, thus, enabling transportation of glass waste with different colors on the same truck without mixing the colors on the tour. This gives rise to the question how the tours of the trucks should be designed given the availability of a device of this kind.

The problem under discussion can be classified as a multi-compartment vehicle routing problem (MCVRP). However, it is different from the ones previously discussed in the literature with respect to the following properties:

- The size of each compartment is not fixed in advance but can be determined individually for each vehicle/each tour.
- The size of the compartments can only be varied discretely, i.e. the walls separating the compartments from each other can only be introduced in specific, predefined positions.
- The number of compartments, into which the capacity of a vehicle is divided, can be identical to the number of product types (glass waste types) but can also be smaller.

Consequently, not only the vehicle tours have to be determined, but it has also to be decided for each vehicle/tour (i) into how many compartments the vehicle capacity should be divided, (ii) what the size of each compartment should be, and (iii) which product type should be assigned to each compartment.

The problem is NP-hard, since it is a generalization of the CVRP (see, for example, Toth & Vigo, 2014). By application of a mathematical model-based exact solution approach, we were only able to solve problem instances with a limited size to optimality. Therefore, a heuristic, namely a variable neighborhood search (VNS), has been developed and will be presented. According to the best of our

* Corresponding author. Tel.: +49 3916711840.
E-mail addresses: tino.henke@ovgu.de (T. Henke), speranza@eco.unibs.it (M.G. Speranza), gerhard.waescher@ovgu.de (G. Wäscher).

http://dx.doi.org/10.1016/j.ejor.2015.05.020
0377-2217/

knowledge, this is the first method which has been proposed for this problem so far. We will further analyze what the economic benefits are which stem from the introduction of flexibly sizable compartments.

The remainder of this paper is organized as follows. Section 2 presents a formal definition and a mathematical formulation of the problem. The relevant literature related to the MCVRP is discussed in Section 3. In Section 4, the proposed variable neighborhood search algorithm is introduced. Extensive numerical experiments have been performed in order to evaluate the mathematical model and the VNS. The design of these experiments and the corresponding results are presented in Section 5. Finally, the main findings are summarized and an outlook on future research is given in Section 6.

2. Problem description and formulation

The multi-compartment vehicle routing problem with flexible compartment sizes (MCVRP-FCS) can be formulated as follows: Let an undirected, weighted graph $G = (V, E)$ be given which consists of a vertex set $V = \{0, 1, \dots, n\}$, representing the location of the depot ($\{0\}$) and the locations of n customers ($\{1, \dots, n\}$), and an edge set $E = \{(i, j): i, j \in V, i < j\}$, representing the edges which can be traveled between the different locations. To each of these edges, a non-negative cost c_{ij}, $(i, j) \in E$, is assigned. It is assumed that all of these costs satisfy the triangle inequality.

Further, let a set P of product types be given. At each vertex (except for the depot) exists a non-negative supply $s_{ip} (i \in V\backslash\{0\}, p \in P)$ of each of the product types. The supplies have to be collected at their locations and transported to the depot without the product types being mixed. A location may be visited several times in order to pick up different product types. However, if being picked up, each supply has to be loaded in total. In other words, a split collection of a single supply is not permitted.

For the purpose of transportation, a set K of homogeneous vehicles is available, each equipped with a total capacity Q. Individually for each vehicle $k \in K$, the total capacity Q can be divided into a limited number $\hat{m}$ of compartments, $\hat{m} \le |P|$, which allows for loading products of different types on a single vehicle while keeping them separated during transportation. The size of the compartments can be varied discretely in equal step sizes, i.e. each compartment size, but also the total vehicle capacity Q, is an integer multiple of a basic compartment unit size q^{unit}. Let the set of these multiples be denoted by $M = \{0, 1, 2, \dots, m^{\max}\}$ where $m^{\max} = Q/q^{\text{unit}}$. Then $q_m = \frac{1}{m^{\max}} \cdot m$, $m \in M$, denotes a compartment size relative to the total capacity Q which consists of m ($m \in M$) multiples of the basic compartment unit size q^{unit}. To illustrate these aspects, we introduce a small example in which the vehicle capacity Q amounts to 200 units and the basic compartment unit size q^{unit} to 10 units. Hence, only compartment sizes of 10, 20, 30,…, 200 units or 5 percent, 10 percent, 15 percent,…, 100 percent of the vehicle capacity can be selected. Accordingly, $m^{\max}$ is equal to 20 and a compartment with $m = 7$ corresponds to a relative compartment size of $q_m = \frac{1}{m^{\max}} \cdot m = \frac{1}{20} \cdot 7 = 0.35$, i.e. 35 percent of the vehicle capacity. It is important to note that the set of potential compartment configurations is identical for all vehicles. However, the actual configuration in a particular solution might be different for each vehicle.

What has to be determined is a set of vehicle tours, an assignment of product types to the vehicles and the sizes of the corresponding compartments such that all supplies are collected, that the capacity of none of the used vehicles is exceeded, and that the total cost of all edges to be traveled is minimized.

This problem involves the following partial decisions to be made simultaneously:

- assignment of product types to each of the vehicles
  (this decision determines which product types can be collected by each vehicle);
- determination of the size of each compartment
  (this decision fixes for each vehicle how its total capacity is split into compartments);
- assignment of supplies to each of the vehicles
  (this decision implicitly includes an assignment of locations to vehicles);
- sequencing of the locations for each of the vehicles
  (this decision determines for each vehicle in which sequence the assigned locations are to be visited).

We note that every vehicle routing problem involves decisions of the last two types, while the first and the second one define the uniqueness of the MCVRP-FCS.

In order to formulate a mathematical model for the MCVRP-FCS, we introduce the following four types of variables:

$$u_{ipk} = \begin{cases} 1, & \text{if supply of product type } p \text{ at location } i \text{ is collected by vehicle } k, \\ 0, & \text{otherwise,} \end{cases} \quad i \in V\backslash\{0\}, p \in P, k \in K;$$

$$x_{ijk} = \begin{cases} 2, & \text{if } i = 0 \text{ and edge } (i, j) \text{ is used twice by vehicle } k, \\ 1, & \text{if edge } (i, j) \text{ is used once by vehicle } k, \\ 0, & \text{otherwise,} \end{cases} \quad i, j \in V : i < j, k \in K;$$

$$y_{pkm} = \begin{cases} 1, & \text{if size } q_m \text{ is selected for product type } p \text{ in vehicle } k, \\ 0, & \text{otherwise,} \end{cases} \quad p \in P, k \in K, m \in M;$$

$$z_{ik} = \begin{cases} 1, & \text{if location } i \text{ is visited by vehicle } k, \\ 0, & \text{otherwise,} \end{cases} \quad i \in V, k \in K.$$

The objective function and the constraints of the model can then be formulated as follows:

$$\min \sum_{(i,j)\in A} \sum_{k\in K} c_{ij} x_{ijk} \tag{1}$$

$$\sum_{k\in K} u_{ipk} = 1 \quad \forall i \in V\backslash\{0\}, p \in P: \ s_{ip} > 0 \tag{2}$$

$$u_{ipk} \le z_{ik} \quad \forall i \in V\backslash\{0\}, p \in P, k \in K \tag{3}$$

$$z_{ik} \le z_{0k} \quad \forall i \in V\backslash\{0\}, k \in K \tag{4}$$

$$\sum_{\substack{j\in V:\\ j>0}} \sum_{k\in K} x_{0jk} \le 2|K| \tag{5}$$

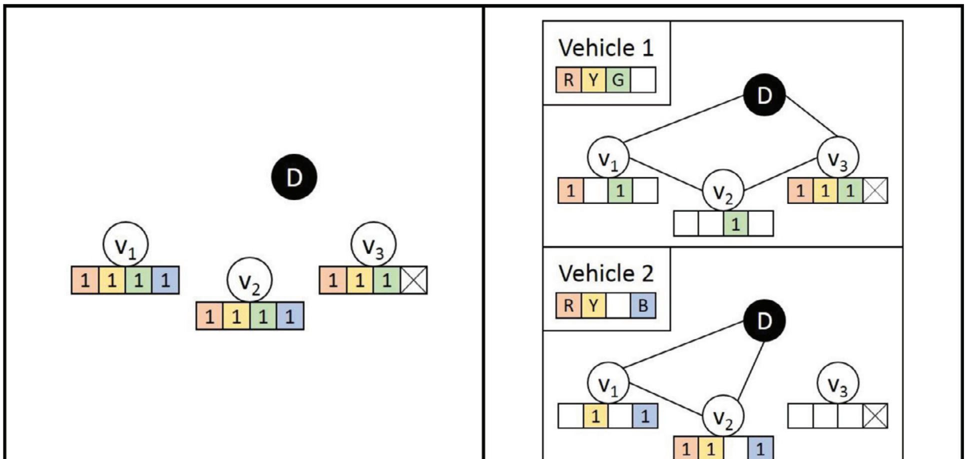
$$\sum_{\substack{j\in V:\\ i<j}} x_{ijk} + \sum_{\substack{j\in V:\\ j<i}} x_{jik} = 2z_{ik} \quad \forall i \in V, k \in K \tag{6}$$

$$\sum_{p\in P} \sum_{m\in M} y_{pkm} \le \hat{m} \quad \forall k \in K \tag{7}$$

$$\sum_{p\in P} \sum_{m\in M} q_m y_{pkm} \le 1 \quad \forall k \in K \tag{8}$$

$$\sum_{i\in V\backslash\{0\}} s_{ip} u_{ipk} \le Q \sum_{m\in M} q_m y_{pkm} \quad \forall p \in P, k \in K \tag{9}$$

$$\sum_{i\in S} \sum_{\substack{j\in S:\\ i<j}} x_{ijk} \le |S| - 1 \quad \forall k \in K, S \subseteq V\backslash\{0\} : |S| > 2 \tag{10}$$



a) A MCVRP-FCS instance with three locations and four product types

b) A solution of the MCVRP-FCS instance

Fig. 1. Illustrative example of a Multi-Compartment Vehicle Routing Problem with Flexible Compartment Sizes.

$$u_{ipk} \in \{0, 1\} \quad \forall i \in V\backslash\{0\}, p \in P, k \in K \tag{11}$$

$$x_{0jk} \in \{0, 1, 2\} \quad \forall j \in V\backslash\{0\}, k \in K \tag{12}$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in V\backslash\{0\}, j \in V\backslash\{0\} : i < j, k \in K \tag{13}$$

$$y_{pkm} \in \{0, 1\} \quad \forall p \in P, k \in K, m \in M \tag{14}$$

$$z_{ik} \in \{0, 1\} \quad \forall i \in V, k \in K \tag{15}$$

The objective function (1) defines the total cost of all tours, which has to be minimized. Constraints (2) guarantee that each positive supply is assigned to exactly one vehicle, while constraints (3) ensure that a supply may only be satisfied by a vehicle if the corresponding customer location is visited by this vehicle. Constraints (4) make sure that the depot is included in each tour. According to constraints (5), the number of tours is limited by the number of available vehicles. Constraints (6) represent the vehicle flow constraints. Constraints (7) ensure that the number of product types assigned to a vehicle may not exceed the maximum number of compartments. Constraints (8) ensure that for each vehicle the sum of the relative compartment sizes q_m must be smaller than or equal to 1 (100 percent), i.e. the sum of the actual compartment sizes must be smaller than or equal to the vehicle capacity. In addition to that, constraints (9) guarantee for each compartment that its capacity is not exceeded by the assigned supplies. Constraints (10) represent the subtour-elimination constraints. Finally, constraints (11)–(15) characterize the variable domains.

An illustrative example for the MCVRP-FCS with three locations (v_1, v_2, v_3) and four product types (R: red, Y: yellow, G: green, B: blue) is provided in Fig. 1a. The locations are depicted as circles while the product types are depicted as squares. The number within a square indicates the amount of supply of a certain product type at a certain location. In case no supply exists, the respective square is crossed. In the shown example, each location has a supply of one unit for each product type with the exception of location 3 not having any supply of product type B at all. Furthermore, each vehicle has a capacity of six units and the maximum number of compartments equals three.

A solution to this problem instance with two vehicles is shown in Fig. 1b. For each vehicle, a distinguished panel is depicted, which consists of information about the product-vehicle-assignment, the supply-vehicle-assignment, and the location sequencing. If a certain product type is assigned to a vehicle, the associated square in the top left part of the panel is colored and named accordingly. Further, if a certain supply is assigned to a vehicle, the corresponding square below the related location is colored and includes the amount of the supply. Finally, the routes are indicated by edges between the vertices.

3. Literature review

Some variants of the MCVRP have been discussed in the literature already. El Fallahi, Prins and Wolfer Calvo (2008) describe a MCVRP arising in the context of distribution of animal food to farms where the sizes of the compartments and the corresponding assignment of product types to the compartments are fixed in advance. They propose a memetic algorithm and a tabu search algorithm for solving this problem. Muyldermans and Pang (2010) consider a similar problem and develop a guided local search algorithm. Furthermore, they identify problem characteristics under which transportation of several product types on vehicles with multiple (a priori fixed) compartments outperforms separate transportation of the different product types on vehicles with a single compartment. Chajakis and Guignard (2003) present a similar problem in the context of deliveries to convenience stores. For their problem, they develop heuristics based on Lagrangian Relaxations. In contrast to these contributions, in this paper a MCVRP with flexible compartment sizes and no a priori assignment of product types to compartments is considered.

Avella, Boccia and Sforza (2004) and Brown and Graves (1981) describe applications of the MCVRP in the context of petrol replenishment. In these problems, the compartment sizes are fixed in advance, while the assignment of product types to compartments is not. In contrast to the MCVRP-FCS, only one supply can be assigned to each compartment, because petrol tanks must be emptied completely during the visit of one customer. Moreover, Cornillier et al. (2008) extend the problem to multiple periods with inventory components.

Derigs et al. (2011) introduce and examine two variants of the MCVRP: The first variant is, in general, similar to the problem described by El Fallahi et al. (2008), Muyldermans and Pang (2010), and Chajakis and Guignard (2003), whereas the second variant is characterized by continuous flexible compartment sizes, while the assignment of product types to compartments is not fixed. However, in contrast to the MCVRP-FCS, each customer may have multiple demands for the same product type. The case in which the number of compartments may be smaller than the number of product types is not explicitly considered. The authors introduce a general mathematical model, i.e. one which covers both problem variants, and a large neighborhood search algorithm which solves their variants of the MCVRP.

A problem similar to the MCVRP-FCS is presented by Caramia and Guerriero (2010). They present a case study-oriented MCVRP with heterogeneous vehicles in milk collection with fixed compartment sizes but no a priori assignment of product types to vehicles. Also, the number of product types (four in the case study) may be smaller than the number of compartments per vehicle (three or more compartments). In contrast to the MCVRP-FCS, they also consider trailers which can be added to a vehicle. For their problem, they present a model-oriented heuristic procedure which assigns demands to vehicles in a first step and determines specific routes for these vehicles in a second step. Only recently, Archetti, Campbell and Speranza (2014) presented theoretical comparisons between multi-commodity and single-commodity vehicle routing problems, where they provide theoretical and empirical insights on relationships between the split delivery vehicle routing problem, the multi-compartment vehicle routing problem, and a combined split delivery and multi-compartment vehicle routing problem. Finally, Mendoza et al. (2010, 2011) discuss the multi-compartment vehicle routing problem with stochastic demands for which they propose several construction procedures and a memetic algorithm.

Apart from the mentioned applications in glass waste collection, fuel delivery, and milk collection, multi-compartment routing problems are found in several other contexts. El Fallahi et al. (2008) describe a problem occurring in the distribution of animal food to farms, in which the food for a certain species should always be assigned to the same compartments because of sanitary rules. Chajakis and Guignard (2003) deal with the simultaneous delivery of dry, refrigerated, and frozen products to convenience stores. Lahyani et al. (2015) present a case study from Tunisia in which olive oil, categorized into different types of quality grades, has to be collected from producers in separate compartments. Finally, multi-compartment problems also occur in the context maritime transportation when bulk products have to be transported (Fagerholt & Christiansen 2000).

Our paper attends to additional aspects which – according to the best of our knowledge – have not been considered in detail in the literature before. These aspects include the case of discretely flexible compartment sizes and the case in which the number of compartments per vehicle is smaller than the number of products types. Especially the second aspect defines a new, unique problem structure for which we suggest a heuristic solution approach.

4. Variable neighborhood search

4.1. Overview

In order to determine good solutions for large problem instances of the MCVRP-FCS, a variable neighborhood search approach with multiple starts (MS-VNS) has been developed. The concept of VNS was first proposed by Mladenović and Hansen (1997) and Hansen and Mladenović (2001). VNS is a local search-based metaheuristic which explores the solution space by means of multiple neighborhood structures.

The general VNS framework can be described as follows: Given a set of different neighborhood structures which are sequenced in a specific order, VNS starts from an initial solution x (also first *incumbent solution*) and randomly selects a neighbor x' of x according to the first neighborhood structure (*shaking*). The neighbor is subsequently improved by application of a local search algorithm, providing a solution x'' (*improvement phase #1*). If this solution does not satisfy a given acceptance criterion, another neighbor x''' of the incumbent x solution is selected randomly, this time, however, from the neighborhood defined by the next neighborhood structure in the sequence, and the above-described process is repeated. Otherwise, if x'' satisfies the acceptance criterion, it is updated as the new incumbent solution, i.e. $x := x''$, and the index of the neighborhood structure is set back to 1, i.e. for the next pass of the improvement phase #1, a neighbor x' of the new incumbent solution of x is selected from the neighborhood defined by the first neighborhood structure. Should no new incumbent solution be determined over a run through all neighborhood structures, the procedure restarts from the first neighborhood structure in the sequence. The procedure continues until a predefined termination criterion is attained. In the following, the execution of a shaking step and the subsequent improvement phase will be referred to as one *iteration*.

Usually the sequence of the neighborhood structures is chosen in a way that the size of the neighborhoods increases as the VNS proceeds. As a consequence, initially the procedure explores neighbors from the incumbent solution, which can be obtained by small changes, while it only moves on to explore larger changes if the search for better solutions was unsuccessful in such “close” neighborhoods.

As we have mentioned before, an important aspect of the MCVRP-FCS are the four different types of partial decisions which have to be made (see Section 2). The usage of a single neighborhood structure may revise several of these decisions simultaneously. However, the combination of decisions is often identical. By the usage of more than one neighborhood structure, VNS is able to revise different combinations of problem decisions, which permits a more diverse exploration of the solution space.

One of the disadvantages of this algorithm is that it often tends to get stuck in a local optimum. In order to overcome this drawback, we implemented our VNS in connection with a multi-start approach. The VNS restarts several times from different initial solutions, which are generated by a randomized construction procedure. The number of starts (in the following called *loops*) is also limited by a termination criterion (*ms termination criterion*). Furthermore, each time a new best solution has been found, it is attempted to improve this solution by a second improvement phase (*improvement phase #2*).

A pseudo-code of our VNS procedure is presented in Fig. 2. Z^{best} denotes the objective function value of the best solution found over all loops and $Z(x)$ denotes the objective function value of an arbitrary solution x. Details of the randomized construction procedure, the solution space, the neighborhood structures and improvement phases, and on the acceptance and termination criteria will be explained in detail in the following subsections.

4.2. A randomized construction procedure for the generation of initial solutions

Each initial solution is generated by means of the following randomized construction procedure. At the beginning, all (positive) supplies are sequenced randomly. According to this sequence, one supply after another is assigned to a vehicle. Starting from the vehicle with the lowest index, it is checked whether the supply can (still) be accommodated by the respective vehicle. This is the case if (i) a compartment has already been opened for the respective product type (because a supply of this product type has already been assigned to the vehicle) or it still can be opened for this product type (because there is still enough vehicle capacity available and the maximum number of compartments is not exceeded) and (ii) the size of the compartment is at least as large as the supply. If the conditions (i) and (ii) hold, then the supply is assigned to the vehicle; otherwise the next vehicle will be checked. The procedure stops when all supplies have been assigned. We remark that the assignment of supplies to vehicles also constitutes an assignment of locations to vehicles. The latter is used as a basis for the determination of an initial set of routes by application of the Lin–Kernighan Heuristic (Lin & Kernighan, 1973).

We further note that the described procedure of assigning supplies to vehicles may result in an assignment, which contains more

```
input: problem data, number of neighborhood structures k^max;

Z^best:= ∞;
do
  generate an initial solution x with objective function value Z(x) randomly;
  set neighborhood structure index k to k:= 1;
  do
    select a neighbor x' from neighborhood structure k of x randomly (shaking);
    apply local search to x' and determine a solution x'' (improvement phase #1);
    if Z(x'') satisfies the acceptance criterion then
      x:= x''; k:= 1;
      if Z(x) < Z^best then
        x^best:= x;
        apply local search procedure to x^best (improvement phase #2);
        Z^best:= Z(x^best);
      endif
    else
      k:= k + 1;
      if k = k^max + 1 then
        k:= 1;
      endif
    endif
  until vns termination criterion is satisfied
until ms termination criterion is satisfied
output: x^best;
```

Fig. 2. Pseudo-code of a VNS for the MCVRP-FCS.

vehicles than actually available. In this case, a large penalty is added to the objective function for each additional vehicle. Consequently, the algorithm is directed to finding solutions with a feasible number of vehicles during the improvement phases. Once a solution with a feasible number of vehicles has been found, then only solutions feasible with respect to the number of available vehicles will be generated.

4.3. Solution space

Throughout the search of the solution space, not only feasible solutions but also solutions which are infeasible with regard to the vehicle capacity can be accepted as incumbent solutions. In order to deal with these infeasibilities, a penalty term is added to the total cost of a solution within the objective function (Gendreau, Hertz and Laporte, 1994). This penalty term is dependent on the extent according to which the vehicle capacities are violated (excess capacity utilization) and a penalty factor α. For the determination of the excess capacity utilization Q_k' of a vehicle k, the selected compartment sizes are considered instead of the actual supplies transported in these compartments:

$$Q_k' = \max\left(0; \sum_{p\in P}\sum_{m\in M} q_m y_{pkm} - Q\right)$$

The modified objective function value $\tilde{Z}_0$ is then obtained by adding the weighted sum of the total excess capacity utilization of all vehicles to the original objective function value Z_0:

$$\tilde{Z}_0 = Z_0 + \alpha \sum_{k\in K} Q'_k$$

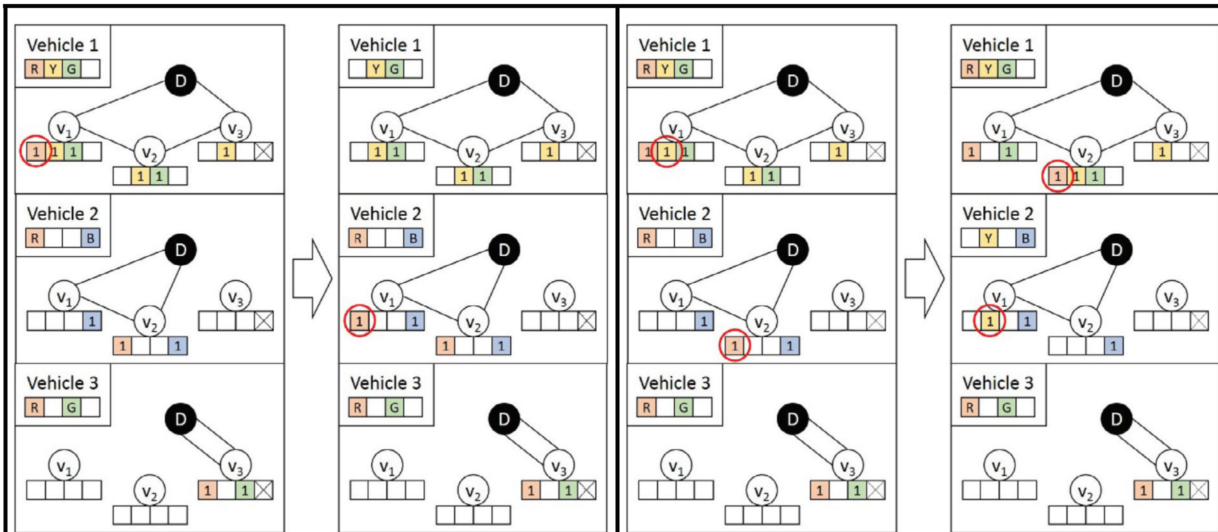
The value of the penalty factor α is adjusted dynamically. At the beginning of each loop, the factor is set to an identical initial value. If no feasible solution was determined for a certain number of consecutive iterations, the factor is increased. If a feasible solution has eventually been found, the factor is reset to its initial value. In this manner, infeasibilities get penalized only lightly during the early stages of the application of the algorithm. However, the longer the solutions remain infeasible, the stronger the penalization gets. Consequently, the algorithm is directed to move to a feasible region of the solution space.

4.4. Neighborhood structures

The crucial part of the design of a VNS approach consists in defining the neighborhood structures from which the solutions are to be selected randomly in the shaking step. Ten different neighborhood structures have been implemented in order to deal with the various decisions, which have to be made when solving the MCVRP-FCS. They can be distinguished into supply-related, location-related, and product type-related neighborhood structures, focusing on changes of supply-vehicle-assignments, location-vehicle-assignments, and product-vehicle-assignments, respectively.

Two supply-related neighborhood structures are used, of which the first one is based on supply shifts, while the second one is based on supply swaps. A supply shift selects a single supply randomly, deletes it from its current vehicle and inserts it into another, randomly selected, vehicle. A supply swap selects two individual supplies randomly from two different vehicles and exchanges both supplies. Through application of these and all following neighborhood structures, it might also be necessary to adjust routes. These adjustments are performed in the following ways: (1) In case all supplies at a location are eliminated from one vehicle, this location is deleted from the corresponding route and the edge between the locations which were adjacent to the eliminated location is added; (2) in case a location has to be added to a vehicle, it is inserted at a random position in the respective route. For both neighborhood structures an example is shown in Fig. 3. Both examples are based on the problem instance shown in Section 2. For the purpose of illustration, three vehicles are used in the initial solution instead of two vehicles as in Fig. 1.

Four different location-related neighborhood structures have been implemented, based on complete location shifts, partial location shifts, location splits, and location swaps. For a complete location shift, one location-vehicle-assignment from the incumbent solution is randomly selected. Then, all supplies related to this specific location, which are currently assigned to the vehicle, are moved collectively to another vehicle. In a partial location shift, only a



a) Supply shift: The supply of product type R at location 1 is moved from vehicle 1 to vehicle 2.

b) Supply swap: The supply of product type Y at location 1 and the supply of product type R at location 2 are exchanged between vehicles 1 and 2.

Fig. 3. Illustration of supply-related neighborhood structures.

randomly determined subset of these supplies is shifted. A location split operator works in a similar way as a complete location shift operator except for the difference that each supply may be shifted to a different vehicle. In a location swap, two location-vehicle-assignments are selected randomly and the corresponding supplies of the locations are exchanged between the vehicles. Pretests have also shown that it is sensible to change more than one location-vehicle-assignment simultaneously, e.g. for a complete location shift not only one location but two locations are shifted to other vehicles. The four described location-related neighborhood structures are illustrated in Fig. 4.

Four different options for the product type-related neighborhood structures have been implemented. Analogously to the location-related neighborhood structures, they are established by complete product shifts, partial product shifts, product splits, and product swaps. In contrast to location-related operators, product-vehicle-assignments instead of location-vehicle-assignments are selected randomly. As for a product shift, the supplies of the selected product type are moved either completely (complete shift) or partially (partial shift) to another vehicle. For the product split, the supplies may be assigned to different vehicles. In a product swap, two product-vehicle-assignments are selected randomly and the corresponding total supplies are exchanged between the two vehicles. An illustration of the product-related neighborhood structures is shown in Fig. 5.

In the VNS algorithm for the MCVRP-FCS, the described neighborhood structures have been implemented in the following sequence: demand shift, demand swap, partial location shift, complete location shift, location swap, location split, partial product shift, complete product shift, product swap, and product split. Only such moves are considered which neither violate the maximum number of product types per vehicle nor the number of available vehicles.

4.5. Improvement phases

During the improvement phase immediately following the shaking step (improvement phase #1), all routes of a solution which have been changed by the shaking step are improved by a local search procedure. This procedure uses the 3-edge neighborhood structure proposed by Lin (1965), where the neighborhood is established by an operator, according to which three edges are eliminated from a route and three new edges are inserted. The search applies the first improvement principle, i.e. the first route is accepted from a neighborhood which has smaller route costs than the current one. The improvement phase terminates when no further improved route can be identified in the neighborhood of the currently considered route.

In the improvement phase, which follows the identification of a new best solution (improvement phase #2), the well-known Lin–Kernighan Heuristic (Lin & Kernighan, 1973) is applied to each of the routes in the solution. For our implementation we used the code provided by Helsgaun (2000).

4.6. Acceptance criterion

VNS algorithms tend to get stuck in local optima (cf. Hansen & Mladenović, 2001). In order to avoid this drawback, we do not only use several starts of the heuristic but also an acceptance criterion which allows for accepting solutions with objective function values worse than those of the best solution found in a loop. The acceptance criterion used for this algorithm is based on threshold accepting (Dueck & Scheuer, 1990) with a dynamically self-adjusting threshold factor.

The dynamic self-adjustment principle is similar to the adjustment procedure for the penalty factor described in Section 4.3. The factor starts with an initial value of 1. After a certain number of consecutive iterations in which no new incumbent solution was accepted, this factor is increased by a certain value. If a solution is eventually accepted as incumbent solution, the factor is reset to 1 again. In this way, the threshold increases slowly if the algorithm is not able to find a better solution in the region of the solution space which is currently explored, enabling the algorithm to overcome local optima.

4.7. Termination criteria

A loop terminates after a certain number of iterations without finding a new best solution. The algorithm terminates after a certain number of loops without finding a new globally best solution.

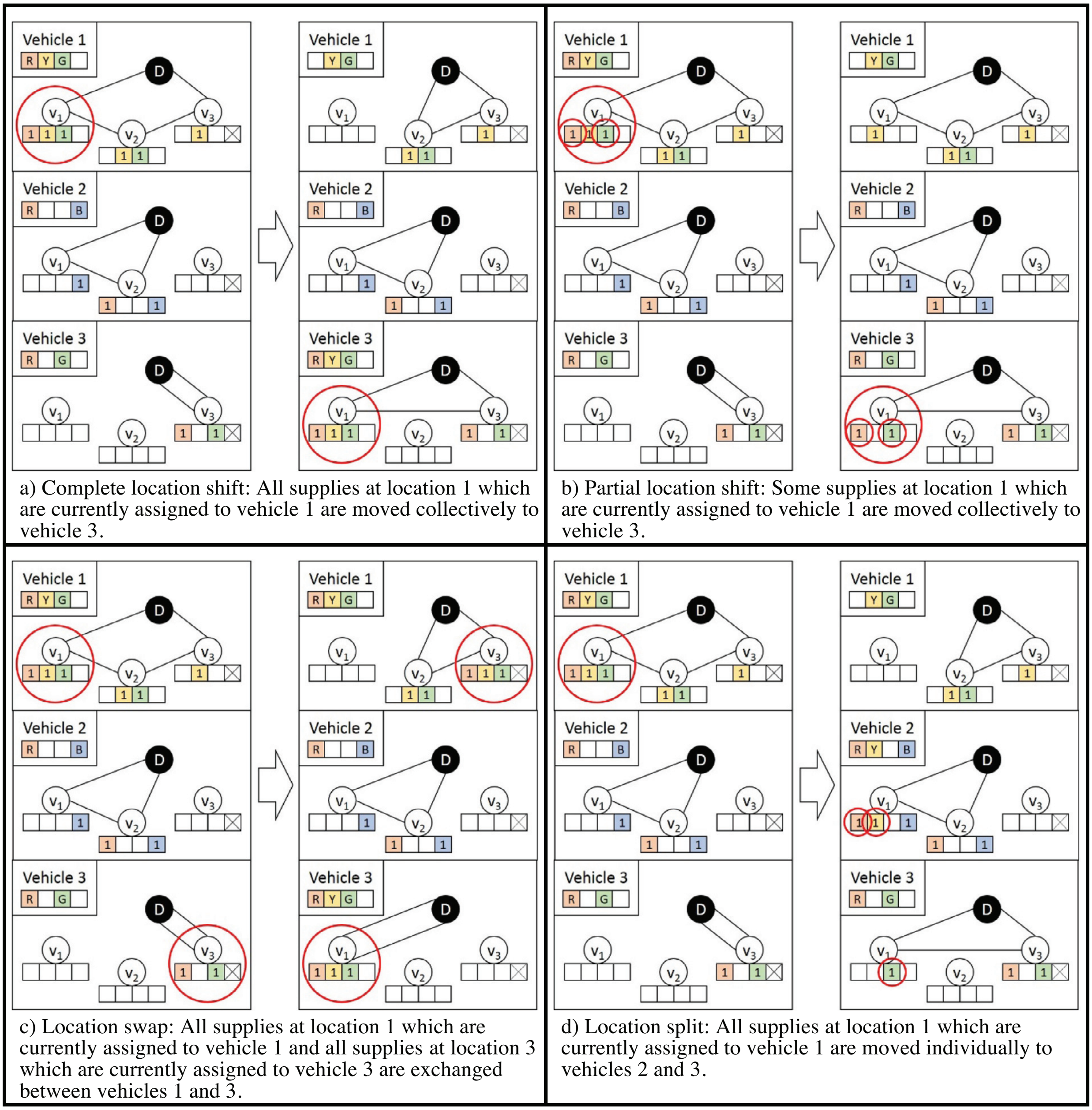


Fig. 4. Illustration of location-related neighborhood structures.

5. Numerical experiments

5.1. Overview

The above-presented model of the MCVRP-FCS was implemented – with a separation procedure for the subtour-elimination constraints – in Microsoft Visual C++ 2012 using CPLEX 12.5. For the MS-VNS algorithm, Microsoft Visual C++ 2012 has been used.

Three sets of numerical experiments were performed in order to evaluate the two approaches. In the first set, the exact solution approach and the VNS algorithm have been applied to relatively small, randomly-generated problem instances. These experiments serve two purposes. Firstly, they provide an impression of the limits of the problem size up to which the MCVRP-FCS can be solved optimally by means of a standard LP solver in reasonable computing time. Secondly, by comparing the results from the MS-VNS approach to the results obtained by the exact approach, the solution quality of the MS-VNS approach can be assessed.

In a second set of experiments, large, randomly-generated instances of the MCVRP-FCS were considered, to which the MS-VNS algorithm was applied. These experiments have been performed in order to study the behavior of the algorithm in greater detail. They were meant to provide insights into how different problem parameters affect solution quality and computing times and, in particular, which parameters make the problem difficult to solve.

In the third set of experiments, the MS-VNS algorithm was applied to problem instances from practice. These experiments were meant to assess the benefits from introducing vehicles with compartments of flexible size in comparison to the utilization of vehicles with a single compartment only.

All experiments were performed on a 3.2 gigahertz and 8 gigabytes RAM personal computer.

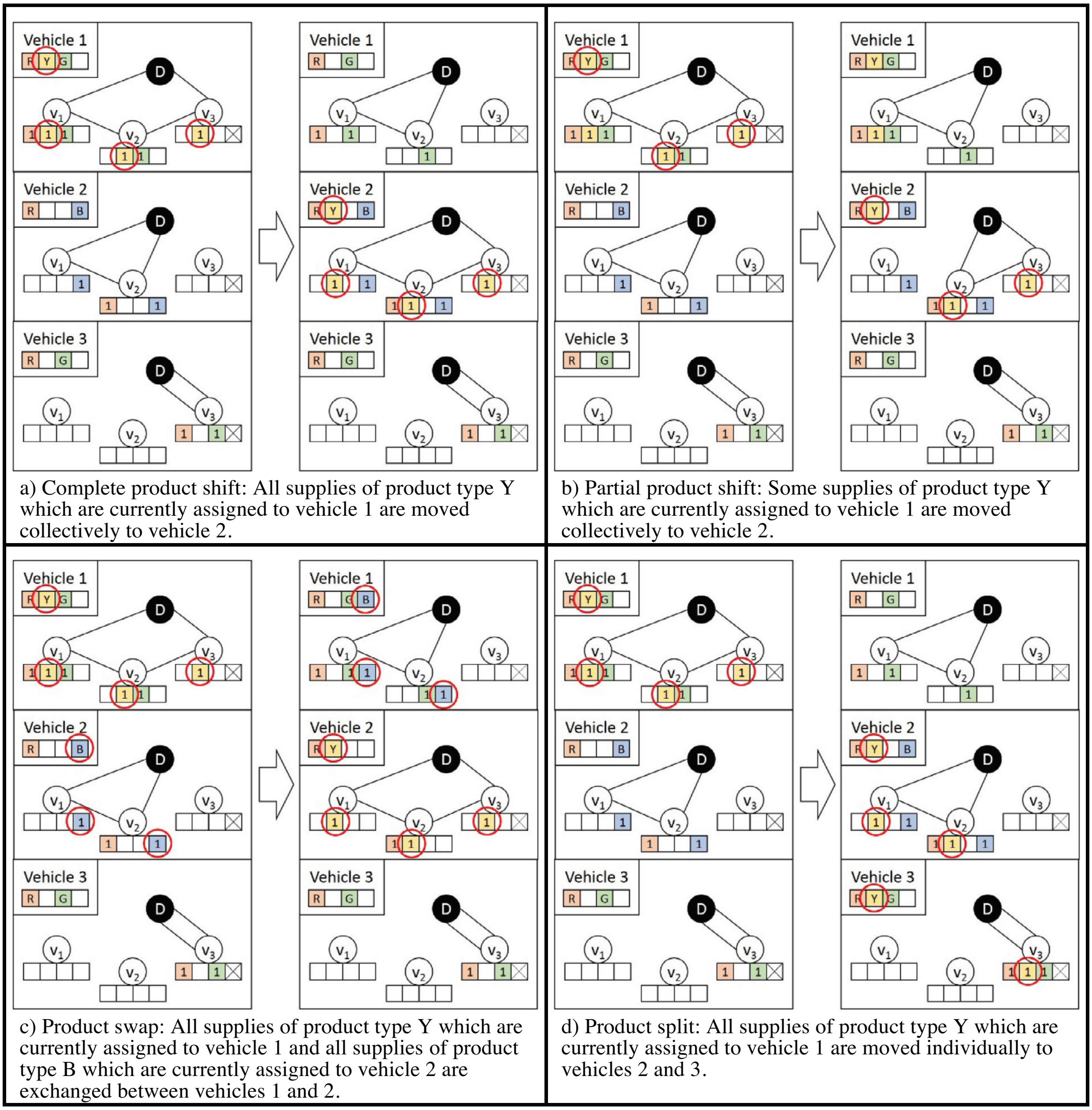


Fig. 5. Illustration of product-related neighborhood structures.

5.2. Problem generation

As we have explained above, the MCVRP-FCS is different from other multi-compartment vehicle routing problems described in the literature so far. Therefore, we could not use benchmark instances from the literature in our experiments. Instead, we randomly generated new problem instances.

In order to establish different classes of problem instances, the parameters of the MCVRP-FCS have been dealt with in the following way. The number of locations n, the number of product types $|P|$, and the number of compartments $\hat{m}$ have been taken as controllable parameters. The respective values chosen for the experiments are given below. Concerning the number of product types of which supplies are available at each location (number of supplies), three options for a supply parameter $\bar{s}$ were considered: small, medium, and large. A small number of supplies corresponds to a situation in which each location provides supplies for at least one product type and at most for one third of the product types ($\bar{s} = 1$, small number of supplies). A medium number of supplies represents a situation where supplies of at least more than one third and at most two thirds of the product types are available at each location ($\bar{s} = 2$, medium number of supplies). Finally, for a large number of supplies, supplies are available at each location of at least more than two thirds of the product types up to $|P|$ (i.e. all) product types ($\bar{s} = 3$, large number of supplies).

The actual supplies for all product types at all locations s_{ip} were generated by means of the following procedure. At first, an initial total supply (S^{Total}) for all product types across all locations was randomly generated from the integer interval $[125 \cdot n; 175 \cdot n]$. This total supply was then split into individual supplies at the locations in several steps. Firstly, based on the respective value of the parameter $\bar{s}$, a number for the supplied product types ($\bar{s}_i$, $i \in V\backslash\{0\}$) was generated randomly for each location i from the corresponding parameter interval for $\bar{s}$. In order to determine which specific product types at a specific location have an actual supply, i.e. $s_{ip} > 0$, a random number out of the interval [0, 0.5] for the first third of product types, [0, 0.3] for the second third of product types, and [0, 0.2] for the last third of product types was drawn. For each location i, these random numbers were sorted in decreasing order, whereas the first $\bar{s}_i$ product types in this sequence were selected to have a supply greater than zero. Then, the total supply quantities for each product type (S_p^{Total}, $p \in P$) were determined in such a manner that 50 percent of the total supply S^{Total} was assigned to the first third of the product types, 30 percent of the total supply to the second third, and 20 percent of the total supply to the last third of the product types. Finally, the actual supplies (s_{ip}, $i \in V \backslash \{0\}$, $p \in P$, $s_{ip} > 0$) were generated by drawing a random number for each supply, which were subsequently normalized according to the total supply of the corresponding product type S_p^{Total}. The biased distribution of product type supplies was used in order to generate realistic instances, since in the glass collection problem the different glass types are more and less frequently disposed.

Finally, the vehicle capacity was set to 1000 for each instance and the number of vehicles was determined by solving a corresponding bin-packing problem exactly in order to guarantee a feasible solution to the problem instance.

All instances can be found online at http://www.mansci.ovgu.de/mansci/en/Research/Materials/2014-p-394.html.

5.3. Experiments with small, randomly-generated instances

The following parameter values were used for the generation of instances for the first set of experiments:

- number of customer locations $n = 10$;
- number of products types $|P| \in \{3, 6, 9\}$;
- number of compartments $\hat{m} \in \{2, 3\}$ for $|P| = 3$, $\hat{m} \in \{2, 4, 6\}$ for $|P| = 6$, $\hat{m} \in \{2, 4, 7, 9\}$ for $|P| = 9$;
- supply parameter $\bar{s} \in \{1, 2, 3\}$.

The number of locations considered in this set of experiments may appear to be small at first sight, but it has to be emphasized that, apart from the number of locations, also the total number of supplies determines the problem size. Since the number of different supplies at each location and the maximum number of compartments per vehicle make the MCVRP-FCS special, we have chosen to explore these parameters in the first place at the expense of considering not more than 10 customer locations. This confinement was necessary in order to limit the respective optimization models to sizes which were still manageable by CPLEX.

The parameter settings resulted in 27 problem classes, for each of which 50 instances have been generated. To all 1350 instances, the exact approach and the MS-VNS algorithm have been applied. $500n|P|$ iterations without improvement (vns termination criterion; cf. Fig. 2) and $(n|P|)/6$ loops without improvement (ms termination criterion) were used as termination criteria for the MS-VNS algorithm. For each problem class, Table 1 presents the number of instances, which have been solved to a proven optimum by the exact approach (#opt.exact) and by the MS-VNS algorithm (#opt.vns), the average objective function value (total cost) per problem instance obtained by the exact approach (tc.exact) and by the MS-VNS algorithm (tc.vns), and computing times per problem instance in seconds for the exact approach (cpu.exact) and the MS-VNS algorithm (cpu.vns). For the exact approach, also the maximal computing time (cpu.exactmax) needed for an instance is reported for each problem class. Furthermore, the average per-instance deviation (aver.gap) of the objective function value obtained by the MS-VNS algorithm from the objective function value obtained by the exact approach, and the corresponding maximum deviation (max.gap) across all instances in a class are presented.

The exact approach managed to solve all of the 1350 instances with a maximal computing time of 21 hours. It appears that the MCVR-FCS becomes more difficult to solve (i.e. the respective computing times increase) with an increasing number of product types, with an increasing number of supplies and with a decreasing number of compartments. Especially the latter observation demonstrates that the specific (partial) decision of this problem, namely the assignment of product types to vehicles, has a significant impact on the complexity of the problem and makes it different from other vehicle routing problems. We further note that – with respect to the computing times – the exact approach presented here cannot be expected to be competitive for large, real-world instances.

Regarding the solution quality, it can be observed that the MS-VNS algorithm performs very well on the instances from these problem classes. From the 1350 instances, the MS-VNS algorithm solves 1334 (98.8 percent) optimally. The average deviation from the optimal objective function value (across all instances) only amounted to 0.02 percent. This is particularly remarkable since the average computing time per problem instance needed by the MS-VNS algorithm (12.97 seconds) represents only 3.0 percent of the average computing time needed by the exact approach (431.31 seconds). Regarding the maximal computing times, this observation is even more considerable, since the maximal computing time for the exact approach is 21 hours whereas the maximal computing time for the MS-VNS is only 53 seconds.

As for the computing times of the MS-VNS algorithm, it can be observed that they increase with the number of product types and the number of demands. In contrast to the exact approach, the impact of the maximum number of compartments on the computing time is inconsistent. For instances with three or six product types, no pattern is observable; for instances with nine product types, the computing times decrease with increasing number of available compartments.

5.4. Experiments with large, randomly-generated instances

For the second set of experiments, a set of large instances was generated. These instances consider 50 locations and up to 9 product types, which results in a maximum of 450 demands. For the number of product types, the number of compartments, and the number of demands, the same combinations of parameters as in the first set of experiments were used. For each combination, one instance was randomly generated. Based on these experiments, the solution quality of the MS-VNS was evaluated.

Since we were neither able to use benchmarks from the literature nor to solve any large instance to optimality, we adopted the following approach. Each instance was solved for 360 minutes by the MS-VNS. The best solutions obtained after 10, 20, 30, 40, 50, and 60, 120, 180, 240, 300, and 360 minutes were recorded and the respective objective function values were compared to the best solution found after 360 minutes. Again, $500n|P|$ iterations without improvement per loop were used.

Table 2 shows for each instance the total costs of the solutions found (tc) during the first hour of computation in steps of 10 minutes and the corresponding deviations to the best known solution (gap w.r.t. BKS). Moreover, Table 3 lists similar results for the 6 hours of computation in steps of 60 minutes. For each instance, the objective function value of the best known solution is indicated by BKS in both tables.

Table 2 shows that, on average, solutions found after 10 minutes have an objective function value which deviates 5.62 percent from the best solution found after 360 minutes. After 60 minutes of running time, this gap is decreased to 2.03 percent on average. Furthermore, one can observe a decrease in the change of the gaps: Whereas the solutions were improved by 1.63 percent and 1.04 percent on average during the second and third 10-minute intervals, respectively, this change decreased to 0.19 percent, 0.27 percent, and 0.45 percent for the remaining three 10-minute intervals. Similar

Table 1
Results for small instances.

Parameters			cplex				vns			gap w.r.t. opt	
$\|P\|$	$\hat{m}$	$\bar{s}$	#opt.exact	tc.exact	cpu.exact	cpu.exactmax	#opt.vns	tc.vns	cpu.vns	avg.gap (percent)	max.gap (percent)
3	2	1 (small)	50	366.9	5.43	52.61	50	366.9	1.19	0.00	0.00
		2 (medium)	50	465.3	11.43	194.87	50	465.35	1.54	0.00	0.00
		3 (large)	50	596.2	101.33	4686.80	50	596.16	2.02	0.00	0.00
	3	1 (small)	50	349.1	2.60	25.84	50	349.08	1.28	0.00	0.00
		2 (medium)	50	337.1	2.66	33.65	50	337.15	1.64	0.00	0.00
		3 (large)	50	336.6	5.00	94.45	50	336.64	1.89	0.00	0.00
6	2	1 (small)	50	550.0	5.91	28.53	50	549.98	3.80	0.00	0.00
		2 (medium)	50	767.5	4.83	21.26	50	767.48	5.47	0.00	0.00
		3 (large)	50	919.8	1.81	3.96	50	919.78	6.44	0.00	0.00
	4	1 (small)	50	367.0	5.04	47.89	50	366.98	5.58	0.00	0.00
		2 (medium)	50	492.1	27.79	120.60	49	492.13	7.60	0.00	0.21
		3 (large)	50	583.8	326.15	2895.40	49	584.46	9.94	0.15	7.39
	6	1 (small)	50	352.0	4.05	72.42	50	351.98	6.03	0.00	0.00
		2 (medium)	50	358.0	5.94	27.14	50	357.98	7.73	0.00	0.00
		3 (large)	50	357.4	8.96	62.29	50	357.42	8.44	0.00	0.00
9	2	1 (small)	50	724.2	502.44	13866.00	49	724.37	18.00	0.03	1.39
		2 (medium)	50	1114.3	1921.59	12333.90	50	1114.25	25.68	0.00	0.00
		3 (large)	50	1419.1	5400.85	76049.40	50	1419.09	36.19	0.00	0.00
	4	1 (small)	50	463.0	12.87	76.28	50	463.05	15.67	0.00	0.00
		2 (medium)	50	647.6	267.69	4653.76	50	647.57	22.87	0.00	0.00
		3 (large)	50	815.3	1324.58	4712.93	50	815.27	26.46	0.00	0.00
	7	1 (small)	50	360.2	5.25	60.90	50	360.16	17.47	0.00	0.00
		2 (medium)	50	456.3	93.21	2239.98	44	457.40	22.15	0.22	6.95
		3 (large)	50	564.5	1550.47	10854.80	46	565.32	30.00	0.15	3.22
	9	1 (small)	50	355.0	3.94	18.95	50	354.98	17.65	0.00	0.00
		2 (medium)	50	392.5	10.53	74.96	50	392.48	22.54	0.00	0.00
		3 (large)	50	383.4	32.94	459.02	47	383.81	25.01	0.12	4.04
Minimum			50	336.6	1.81	3.96	44	336.64	1.19	0.00	0.00
Average			50	551.6	431.31	4954.39	49.4	551.75	12.97	0.02	0.86
Maximum			50	1419.1	5400.85	76049.40	50	1419.09	36.19	0.22	7.39

Table 2
Results for large instances, 1 hour of computing time.

$\|P\|$	$\hat{m}$	$\bar{s}$	Best solution found after												BKS
			10 minutes		20 minutes		30 minutes		40 minutes		50 minutes		60 minutes		
			tc	gap w.r.t. BKS (percent)	tc	gap w.r.t. BKS (percent)	tc	gap w.r.t. BKS (percent)	tc	gap w.r.t. BKS (percent)	tc	gap w.r.t. BKS (percent)	tc	gap w.r.t. BKS (percent)	
3	2	1 (small)	1117.79	3.96	1117.79	3.96	1101.82	2.47	1097.24	2.05	1097.24	2.05	1080.68	0.51	1075.21
		2 (medium)	1136.31	0.00	1136.31	0.00	1136.31	0.00	1136.31	0.00	1136.31	0.00	1136.31	0.00	1136.31
		3 (large)	1529.73	4.84	1459.09	0.00	1459.09	0.00	1459.09	0.00	1459.09	0.00	1459.09	0.00	1459.09
	3	1 (small)	1347.26	6.96	1259.55	0.00	1259.55	0.00	1259.55	0.00	1259.55	0.00	1259.55	0.00	1259.55
		2 (medium)	1792.21	10.89	1792.21	10.89	1667.96	3.20	1667.96	3.20	1667.96	3.20	1667.96	3.20	1616.20
		3 (large)	1830.68	2.75	1830.68	2.75	1830.68	2.75	1830.68	2.75	1830.68	2.75	1830.68	2.75	1781.74
6	2	1 (small)	1323.69	0.00	1323.69	0.00	1323.69	0.00	1323.69	0.00	1323.69	0.00	1323.69	0.00	1323.69
		2 (medium)	1970.85	8.07	1970.85	8.07	1844.73	1.15	1844.73	1.15	1844.73	1.15	1844.73	1.15	1823.69
		3 (large)	2879.94	20.16	2396.85	0.00	2396.85	0.00	2396.85	0.00	2396.85	0.00	2396.85	0.00	2396.85
	4	1 (small)	1112.26	0.00	1112.26	0.00	1112.26	0.00	1112.26	0.00	1112.26	0.00	1112.26	0.00	1112.26
		2 (medium)	1627.56	6.42	1627.56	6.42	1627.56	6.42	1627.56	6.42	1627.56	6.42	1589.01	3.90	1529.42
		3 (large)	2486.69	8.06	2469.11	7.30	2469.11	7.30	2434.65	5.80	2372.35	3.09	2372.35	3.09	2301.20
	6	1 (small)	1648.06	3.48	1647.35	3.43	1631.44	2.44	1631.44	2.44	1631.44	2.44	1631.44	2.44	1592.65
		2 (medium)	2069.92	3.48	2069.92	3.48	2069.92	3.48	2069.92	3.48	2069.92	3.48	2052.98	2.63	2000.30
		3 (large)	2472.03	6.59	2472.03	6.59	2472.03	6.59	2428.83	4.73	2387.18	2.93	2387.18	2.93	2319.24
9	2	1 (small)	1607.41	0.71	1607.41	0.71	1597.10	0.06	1597.10	0.06	1597.10	0.06	1597.10	0.06	1596.12
		2 (medium)	2726.08	2.95	2702.56	2.06	2702.56	2.06	2702.56	2.06	2702.56	2.06	2702.56	2.06	2648.00
		3 (large)	3343.61	7.10	3268.01	4.68	3268.01	4.68	3268.01	4.68	3268.01	4.68	3268.01	4.68	3121.83
	4	1 (small)	1480.58	8.33	1480.58	8.33	1463.77	7.10	1463.77	7.10	1463.47	7.07	1393.81	1.98	1366.79
		2 (medium)	1541.96	0.00	1541.96	0.00	1541.96	0.00	1541.96	0.00	1541.96	0.00	1541.96	0.00	1541.96
		3 (large)	1977.32	0.00	1977.32	0.00	1977.32	0.00	1977.32	0.00	1977.32	0.00	1977.32	0.00	1977.32
	7	1 (small)	1601.54	13.64	1534.57	8.89	1509.99	7.14	1490.15	5.73	1490.15	5.73	1490.15	5.73	1409.34
		2 (medium)	2517.56	7.32	2517.56	7.32	2494.68	6.34	2494.68	6.34	2494.68	6.34	2494.68	6.34	2345.86
		3 (large)	3830.43	6.56	3711.65	3.25	3711.65	3.25	3711.65	3.25	3612.92	0.51	3612.92	0.51	3594.65
	9	1 (small)	1843.51	6.36	1843.51	6.36	1789.81	3.26	1789.81	3.26	1789.81	3.26	1789.81	3.26	1733.29
		2 (medium)	2816.5	7.21	2816.50	7.21	2793.74	6.35	2793.74	6.35	2793.74	6.35	2793.74	6.35	2627.05
		3 (large)	4117.89	5.83	4117.89	5.83	4023.72	3.41	4023.72	3.41	4023.72	3.41	3942.29	1.32	3890.97
Minimum				0.00		0.00		0.00		0.00		0.00		0.00	
Average				5.62		3.98		2.94		2.75		2.48		2.03	
Maximum				20.16		10.89		7.30		7.10		7.07		6.35	

Table 3
Results for large instances, 6 hours of computing time.

$\|P\|$	$\hat{m}$	$\bar{s}$	Best solution found after											
			60 minutes		120 minutes		180 minutes		240 minutes		300 minutes		360 minutes	
			tc	gap w.r.t. BKS (percent)	tc	gap w.r.t. BKS (percent)	tc	gap w.r.t. y (percent)	tc	gap w.r.t. BKS (percent)	tc	gap w.r.t. y (percent)	tc (BKS)	gap w.r.t. BKS (percent)
3	2	1 (small)	1080.68	0.51	1080.68	0.51	1075.76	0.05	1075.21	0.00	1075.21	0.00	1075.21	0.00
		2 (medium)	1136.31	0.00	1136.31	0.00	1136.31	0.00	1136.31	0.00	1136.31	0.00	1136.31	0.00
		3 (large)	1459.09	0.00	1459.09	0.00	1459.09	0.00	1459.09	0.00	1459.09	0.00	1459.09	0.00
	3	1 (small)	1259.55	0.00	1259.55	0.00	1259.55	0.00	1259.55	0.00	1259.55	0.00	1259.55	0.00
		2 (medium)	1667.96	3.20	1616.20	0.00	1616.20	0.00	1616.20	0.00	1616.20	0.00	1616.20	0.00
		3 (large)	1830.68	2.75	1781.74	0.00	1781.74	0.00	1781.74	0.00	1781.74	0.00	1781.74	0.00
6	2	1 (small)	1323.69	0.00	1323.69	0.00	1323.69	0.00	1323.69	0.00	1323.69	0.00	1323.69	0.00
		2 (medium)	1844.73	1.15	1844.73	1.15	1844.73	1.15	1844.73	1.15	1844.73	1.15	1823.69	0.00
		3 (large)	2396.85	0.00	2396.85	0.00	2396.85	0.00	2396.85	0.00	2396.85	0.00	2396.85	0.00
	4	1 (small)	1112.26	0.00	1112.26	0.00	1112.26	0.00	1112.26	0.00	1112.26	0.00	1112.26	0.00
		2 (medium)	1589.01	3.90	1589.01	3.90	1571.95	2.78	1571.95	2.78	1529.42	0.00	1529.42	0.00
		3 (large)	2372.35	3.09	2354.77	2.33	2301.20	0.00	2301.20	0.00	2301.20	0.00	2301.20	0.00
	6	1 (small)	1631.44	2.44	1609.81	1.08	1592.65	0.00	1592.65	0.00	1592.65	0.00	1592.65	0.00
		2 (medium)	2052.98	2.63	2052.98	2.63	2039.76	1.97	2012.06	0.59	2000.30	0.00	2000.30	0.00
		3 (large)	2387.18	2.93	2368.14	2.11	2368.14	2.11	2368.14	2.11	2319.24	0.00	2319.24	0.00
9	2	1 (small)	1597.100	0.06	1596.12	0.00	1596.12	0.00	1596.12	0.00	1596.12	0.00	1596.12	0.00
		2 (medium)	2702.56	2.06	2675.37	1.03	2675.37	1.03	2661.55	0.51	2648.00	0.00	2648.00	0.00
		3 (large)	3268.01	4.68	3268.01	4.68	3248.53	4.06	3121.83	0.00	3121.83	0.00	3121.83	0.00
	4	1 (small)	1393.81	1.98	1393.81	1.98	1393.81	1.98	1393.81	1.98	1393.81	1.98	1366.79	0.00
		2 (medium)	1541.96	0.00	1541.96	0.00	1541.96	0.00	1541.96	0.00	1541.96	0.00	1541.96	0.00
		3 (large)	1977.32	0.00	1977.32	0.00	1977.32	0.00	1977.32	0.00	1977.32	0.00	1977.32	0.00
	7	1 (small)	1490.15	5.73	1472.07	4.45	1409.34	0.00	1409.34	0.00	1409.34	0.00	1409.34	0.00
		2 (medium)	2494.68	6.34	2464.19	5.04	2370.69	1.06	2370.69	1.06	2345.86	0.00	2345.86	0.00
		3 (large)	3612.92	0.51	3612.92	0.51	3612.92	0.51	3612.92	0.51	3594.65	0.00	3594.65	0.00
	9	1 (small)	1789.81	3.26	1765.17	1.84	1759.60	1.52	1741.41	0.47	1741.41	0.47	1733.29	0.00
		2 (medium)	2793.74	6.35	2709.40	3.13	2709.40	3.13	2709.40	3.13	2705.09	2.97	2627.05	0.00
		3 (large)	3942.29	1.32	3942.29	1.32	3890.97	0.00	3890.97	0.00	3890.97	0.00	3890.97	0.00
Minimum				0.00		0.00		0.00		0.00		0.00		0.00
Average				2.03		1.40		0.79		0.53		0.24		0.00
Maximum				6.35		5.04		4.06		3.13		2.97		0.00

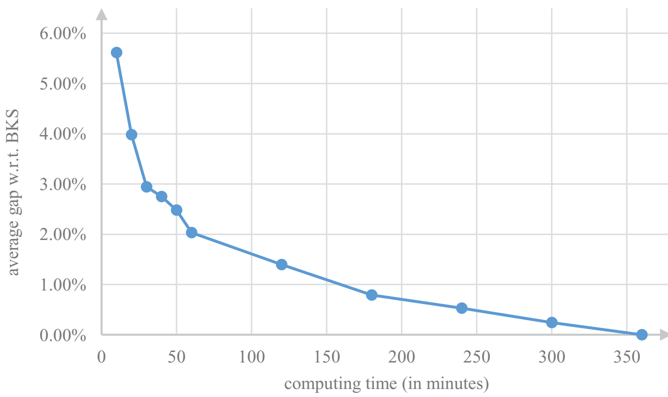


Fig. 6. Average gap with respect to the best known solution plotted against computing time.

results are shown by Table 3: During the second and third 60-minute interval, the average gap is reduced by 0.64 percent and 0.61 percent, respectively. In contrast to this development, the decrease during the last three intervals is only 0.25 percent, 0.29 percent, and 0.24 percent. Fig. 6 illustrates these findings, showing a decreasing average improvement with increasing computing time. Therefore, we cautiously conclude that the objective function values converge. Regarding different problem parameters, no significant impact on the solution quality can be observed. Therefore, the MS-VNS seems to perform with similar quality for all problem settings.

5.5. Experiments with instances from practice

In the third set of experiments, we investigated the benefits of introducing vehicles with multiple compartments over vehicles with a single compartment only. For these experiments, we generated instances based on data from practice, which we obtained from a recycling company based in the City of Magdeburg, Germany. The company is responsible for picking up glass waste from containers, serving 127 containers at 44 different locations. Up to three different containers are available at each location, provided for the collection of colorless, green and brown glass. The trucks, which are used for picking up the glass waste at the container locations, possess a single flexible wall that can be introduced at predefined positions such that the total capacity of the truck can be divided into at most two compartments. The position of the wall can be rearranged for each vehicle tour.

The company provided us with a list which – over a period of several weeks – documented for each day which location was visited, which product type (glass color) was picked up, and what the respective supplies were. From this data we were able to estimate the so-called fill rates, i.e. the amount of glass by which a container is filled per day.

Over a planning horizon of 25 days we then deterministically simulated the development of the supplies for each container at each location, assuming that a container is always emptied on the last day before its capacity would be exceeded. The data (i.e. the supplies for each container at each location) that has been obtained by the simulation for each day has then been considered as the basic data for the definition of a specific routing problem instance. These 25 days have been selected for our numerical experiments, giving rise to 25 different problem instances. The distances between the locations and between the depot and the locations were determined by means of a commercial navigation system and taken as the costs of the corresponding edges. It was ensured that the distance matrix satisfies the triangle inequality.

The procedure resulted in instances where the number of container locations varies between 26 and 34 and the number of supplies between 34 and 54. The number of product types is three for all instances. The MS-VNS approach was applied to the 25 instances allowing either one or up to two compartments for each vehicle. For both scenarios, all algorithm settings were identical. As termination criteria, $500n|P|$ iterations without improvement and $n|P|/6$ loops without improvement were used. The average computing times amounted to 261 seconds for the CVRP-scenario and 452 seconds for the MCVRP-scenario.

Table 4 presents the characteristics of each problem instance and the corresponding results from the experiment, namely the number of locations (n) and the total number of supplies ($\hat{s}$), the total costs of the solutions in case only one compartment can be used on each vehicle (tc.cvrp), the total costs of the solutions if up to two (flexible-sized) compartments can be introduced (tc.mcvrp) and the improvement (impr; in percent) which can be obtained by the latter. Finally, the corresponding number of tours for vehicles with one compartment (tours.cvrp) and with up to two compartments (tours.mcvrp) are shown.

The results of this set of experiments demonstrate the economic benefits of using vehicles with multiple, flexible-size compartments over vehicles with single compartments. For these problem instances, the costs of the tours could be reduced by 34.8 percent on average. Furthermore, the number of tours (or vehicles) necessary on average to pick up all supplies was reduced by one, from 3.2 tours to 2.0 tours.

6. Conclusions

In this paper, a variant of the multi-compartment vehicle routing problem was introduced and studied. This problem is different from the ones previously discussed in the literature. In particular, it includes compartments of flexible sizes, allowing for the number of compartments being smaller than the number of product types, which have to be transported separately. Because of this aspect, an additional question arises concerning the assignment of product types to vehicles.

The main contributions of this paper include (1) a presentation of this new variant of the MCVRP which is inspired by a real-world application; (2) the development of an exact and a heuristic solution procedure for this problem; and (3) extensive numerical experiments investigating the problem itself as well as the performance of both suggested algorithms.

Concerning the exact approach, the results of the experiments have shown that only problem instances of limited size can be solved to optimality in reasonable computing time. Detailed analysis of the results from these experiments provided insights into the impact of different problem parameters: The problem becomes – not unexpectedly – harder to solve with an increasing number of customer locations, product types and supplies. However, it was also shown that the maximum number of compartments, which determines the number of product types per vehicles, has a significant impact on the complexity of the problem. The smaller the number of available compartments is, the harder the problem becomes to solve. This result particularly emphasizes the relevance of the additional assignment decision encountered in this problem, namely the assignment of product types to be transported by each vehicle.

Concerning the heuristic approach, we were able to show that it performs well – producing optimal or near-optimal solutions for small problem instances. For large instances with up to 450 demands, the heuristic finds presumably good quality solutions in reasonable time.

In a third experiment, based on instances from practice, the benefits of using vehicles with multiple compartments of flexible sizes over using vehicles with a single compartment were investigated. The results for this specific application have shown that – if multiple, flexible-size compartments can be introduced – the total costs of the

Table 4
Results for instances from practice.

Instance	Parameters		Total costs			Number of tours	
	n	$\hat{s}$	tc.cvrp	tc.mcvrp	impr (percent)	tours.cvrp	tours.mcvrp
1	32	54	123.83	83.375	32.7	3	2
2	26	38	117.59	78.054	33.6	3	2
3	33	44	130.18	91.188	30.0	3	2
4	33	46	136.24	95.543	29.9	3	2
5	31	48	149.20	79.975	46.4	4	2
6	29	34	115.05	77.243	32.9	3	2
7	34	54	163.42	94.532	42.2	4	2
8	28	40	132.67	92.671	30.1	3	2
9	30	44	117.10	78.699	32.8	3	2
10	31	44	122.70	82.282	32.9	3	2
11	34	48	160.17	90.26	43.6	4	2
12	30	36	125.60	85.696	31.8	3	2
13	32	54	123.83	83.475	32.6	3	2
14	26	38	117.59	78.054	33.6	3	2
15	33	44	130.18	91.188	30.0	3	2
16	33	46	136.24	95.543	29.9	3	2
17	31	48	149.20	79.975	46.4	4	2
18	29	34	115.05	77.243	32.9	3	2
19	34	54	163.42	94.532	42.2	4	2
20	28	40	132.67	92.671	30.1	3	2
21	30	44	117.10	78.699	32.8	3	2
22	31	44	122.70	82.282	32.9	3	2
23	34	48	160.17	90.26	43.6	4	2
24	30	36	125.60	85.696	31.8	3	2
25	32	54	123.83	83.475	32.6	3	2
Minimum	26	34	115.1	77.2	29.9	3	2
Average	31.0	44.6	132.5	85.7	34.8	3.2	2.0
Maximum	34	54	163.4	95.5	46.4	4	2

tours necessary for collecting all supplies can be reduced drastically. These results indicate the necessity of dealing with and developing effective solutions approaches for multi-compartment vehicle routing problems.

References

Archetti, C., Campbell, A., & Speranza, M. G. (2014). Multi-commodity vs. single-commodity routing. *Transportation Science,* in press. doi:10.1287/trsc.2014.0528.

Avella, P., Boccia, M., & Sforza, A. (2004). Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research, 152*, 170–179.

Brown, G. G., & Graves, G. W. (1981). Real-time dispatch of petroleum tank trucks. *Management Science, 27*, 19–32.

Caramia, M., & Guerriero, F. (2010). A milk collection problem with incompatibility constraints. *Interfaces, 40*, 130–143.

Chajakis, E. D., & Guignard, M. (2003). Scheduling deliveries in vehicles with multiple compartments. *Journal of Global Optimization, 26*, 43–78.

Cornillier, F., Boctor, F. F., Laporte, G., & Renaud, J. (2008). A heuristic for the multi-period petrol station replenishment problem. *European Journal of Operational Research, 191*, 295–305.

Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., & Vogel, U. (2011). Vehicle routing with compartments: Applications, modelling and heuristics. *OR Spectrum 33*, 885–914.

Dueck, G., & Scheuer, T. (1990). Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics, 90*, 161–175.

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

Fagerholt, K., & Christiansen, M. (2000). A combined ship scheduling and allocation problem. *Journal of the Operational Research Society, 51*, 834–842.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science, 40*, 1276–1290.

Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). *The vehicle routing problem: Latest advances and new challenges*. New York: Springer.

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

Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research, 126*, 106–130.

Lahyani, R., Coelho, L. C., Khemakhem, M., Laporte, G., & Semet, F. (2015). A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. *Omega, 51*, 1–10.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science, 43*, 408–416.

Lin, S. (1965). Computer solutions to the traveling salesman problem. *Bell Systems Technical Journal, 44*, 2245–2269.

Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations Research, 21*, 498–516.

Mendoza, J. E., Castanier, B., Guéret, C., Medaglia, A. L., & Velasco, N. (2010). A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research, 37*, 1886–1898.

Mendoza, J. E., Castanier, B., Guéret, C., Medaglia, A. L., & Velasco, N. (2011). Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. *Transportation Science, 45*, 346–363.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research, 24*, 207–226.

Muyldermans, L., & Pang, G. (2010). On the benefits of co-collection: Experiments with a multi-compartment vehicle routing problem. *European Journal of Operational Research, 206*, 93–103.

Toth, P., & Vigo, D. (2014). *Vehicle routing: problems, methods, and applications* (2nd ed.). Philadelphia: Society for Industrial and Applied Mathematics.

IV

A Branch-and-Cut Algorithm for the Multi-Compartment Vehicle Routing Problem with Flexible Compartment Sizes

A Branch-and-Cut Algorithm for the Multi-Compartment Vehicle Routing Problem with Flexible Compartment Sizes

Tino Henke
Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany
tino.henke@ovgu.de, +49-391-6751840

M. Grazia Speranza
Department of Quantitative Methods, University of Brescia, 25122 Brescia, Italy
grazia.speranza@unibs.it

Gerhard Wäscher
Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany
School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, 100044 Beijing, China
gerhard.waescher@ovgu.de

Abstract

Multi-compartment vehicle routing problems arise in a variety of problem settings in which different product types have to be transported separated from each other. In this paper, a problem variant which occurs in the context of glass waste recycling is considered. In this problem, a set of locations exists, each of which offering a number of containers for the collection of different types of glass waste (e.g. colorless, green, brown glass). In order to pick up the contents from the containers, a fleet of homogeneous disposal vehicles is available. Individually for each disposal vehicle, the capacity can be discretely separated into a limited number of compartments to which different glass waste types are assigned. The objective of the problem is to minimize the total distance to be travelled by the disposal vehicles.

For solving this problem to optimality, a branch-and-cut algorithm has been developed and implemented. Extensive numerical experiments have been conducted in order to evaluate the algorithm and to gain insights into the problem structure. The corresponding results show that the algorithm is able to solve instances with up to 50 locations to optimality and that it reduces the computing time by 87% compared to instances from the literature. Additional experiments give managerial insights into the use of different variants of compartments with flexible sizes.

Keywords: vehicle routing, multiple compartments, branch-and-cut algorithm, waste collection

1 Introduction

Vehicle routing problems have been studied for almost 60 years in numerous variants (for surveys see Golden et al. 2008, Laporte 2009, or Toth and Vigo 2014). In recent years, researchers have been especially interested in so-called rich vehicle routing problems which take into account various real-world problem extensions. One group of such rich vehicle routing problems consists of multi-compartment vehicle routing problems. In a multi-compartment vehicle routing problem (MCVRP), several product types are considered which have to be transported separately. This may be necessary because of different transportation requirements, e.g. different temperatures for food products, or because of the consistence of the product types, e.g. liquid or bulk products. Consequently, the vehicles have not only one but several storage compartments in order to serve more than one product type and, thus, to allow for efficient transportation.

A variety of applications of MCVRPs has been mentioned in the literature (see Section 3). These include, amongst others, the distribution of petrol products, the delivery of food products, or the collection of waste. The MCVRP variant considered in this paper is motivated by a real-world problem which occurs in the context of glass waste collection, in which several types of glass waste, i.e. colorless, green and brown glass, have to be collected from public glass waste containers. This problem has the special characteristics that the number of compartments is variable and that the compartment sizes are flexible within a pre-defined set of potential compartment sizes. In the following, this problem will be referred to as the multi-compartment vehicle routing problem with discretely flexible compartment sizes (MCVRP-DFCS).

Although some exact methods have been proposed in order to solve MCVRPs, so far only a single mathematical programming formulation has been presented to solve the MCVRP-DFCS. This paper introduces a new formulation and a branch-and-cut algorithm which includes subtour-elimination cuts, capacity cuts, and several newly developed or adapted valid inequalities. The algorithm has been tested by means of extensive numerical experiments on instances from the literature and newly generated instances. Compared to results from the literature, the computing times can be decreased by 87.19% on average. Furthermore, the experiments demonstrate a good performance of the algorithm for instances with up to 50 locations and give detailed insights into the problem structure. Moreover, the application of different types of flexible compartments has been investigated.

The remainder of this paper is organized in the following manner. In Section 2, the MCVRP-DFCS is described in detail and the new mathematical programming formulation is presented. In Section 3, a brief overview of the existing literature on MCVRPs is given and the differences between the MCVRP-DFCS and other problem variants are identified. Afterwards, the branch-and-cut algorithm with all its components is explained in detail in Section 4. In Section 5, the design of the numerical experiments and the corresponding results are explained and presented, before some conclusions are given in the last section.

2 Problem Description and Formulation

In the following, the MCVRP-DFCS is formulated on an undirected, complete and weighted graph $G = (V, E)$, which consists of a vertex set V and an edge set E. The vertex set $V = \{0,1, \dots, \mathrm{n}\}$ represents the location of the depot $\{0\}$ and the locations of n customers $\{1,2, \dots, \mathrm{n}\}$, whereas the edge set $E = \{(\mathrm{i}, \mathrm{j}): \mathrm{i}, \mathrm{j} \in V, \mathrm{i} < \mathrm{j}\}$ includes all edges which can be travelled between locations. A non-negative cost $\mathrm{c_{ij}}, (\mathrm{i}, \mathrm{j}) \in E$, is assigned to each edge.

Furthermore, a set of different product types P is considered. For each of these product types a non-negative supply $\mathrm{s_{ip}} (\mathrm{i} \in V\backslash\{0\}, \mathrm{p} \in P)$ exists at each customer location. Each supply must be completely collected by exactly one vehicle and has to be transported to the depot. However, it is not necessary to collect supplies of different product types at the same location by the same vehicle.

In order to collect and transport the supplies, a set of homogeneous vehicles K is available at the depot. Individually for each vehicle, its total capacity Q can be separated into compartments, where each compartment allows for the transportation of exactly one product type. The maximal number $\widehat{\mathrm{m}}$ of compartments into which the capacity can be divided is limited and may be smaller than the number of product types. The size of each compartment is flexible and can be selected from a pre-defined set of potential compartment sizes. The smallest feasible compartment size in this set is the basic compartment unit size $\mathrm{q^{unit}}$, and the remaining potential compartments sizes are all integer multiples of $\mathrm{q^{unit}}$ which do not exceed the total capacity.

The objective of the problem is to assign all supplies to vehicles and to determine a set of vehicle routes such that all capacities are respected and the total cost is minimized.

In the MCVRP-DFCS several decisions have to be made simultaneously. For each vehicle, it must be decided which product types are to be assigned to this vehicle and which compartment size should be selected for each assigned product type. Furthermore, for each vehicle it must be decided which supplies are to be collected and, thus, which locations are to be visited. Finally, the sequence according to which the assigned locations are to be visited has to be determined for each vehicle. In order to formulate a mathematical model for the MCVRP-DFCS, the following five types of variables are introduced:

$$\mathrm{u_{ipk}} = \begin{cases} 1, \text{ if supply of product type p at location i is collected by vehicle k,} \\ 0, \text{ otherwise,} \end{cases} \qquad \mathrm{i} \in V\backslash\{0\}, \mathrm{p} \in P, \mathrm{k} \in K;$$

$$\mathrm{x_{ijk}} = \begin{cases} 2, \text{ if i} = 0 \text{ and edge (i,j) is used twice by vehicle k,} \\ 1, \text{ if edge (i,j) is used once by vehicle k,} \\ 0, \text{ otherwise,} \end{cases} \qquad \mathrm{i}, \mathrm{j} \in V, \mathrm{i} < \mathrm{j}, \mathrm{k} \in K;$$

$$\mathrm{z_{ik}} = \begin{cases} 1, \text{ if location i is visited by vehicle k,} \\ 0, \text{ otherwise,} \end{cases} \qquad \mathrm{i} \in V, \mathrm{k} \in K;$$

$$\mathrm{y^B_{pk}} = \begin{cases} 1, \text{ if product type p is assigned to a compartment of vehicle k,} \\ 0, \text{ otherwise,} \end{cases} \qquad \mathrm{p} \in P, \mathrm{k} \in K;$$

$\mathrm{y^I_{pk}}$: Integer variable indicating the size of the compartment of vehicle k which is assigned to product type p in number of basic unit sizes $\mathrm{q^{unit}}$ $\qquad \mathrm{p} \in P, \mathrm{k} \in K.$

The objective function and the constraints of the model can then be formulated as follows:

$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ i<j}} \sum_{k \in K} c_{ij} \cdot x_{ijk} \tag{1}$$

$$\sum_{k \in K} u_{ipk} = 1 \qquad \forall i \in V \setminus \{0\}, p \in P, s_{ip} > 0 \tag{2}$$

$$u_{ipk} \leq z_{ik} \qquad \forall\, i \in V \setminus \{0\}, p \in P, k \in K \tag{3}$$

$$z_{ik} \leq z_{0k} \qquad \forall\, i \in V \setminus \{0\}, k \in K \tag{4}$$

$$\sum_{j \in V \setminus \{0\}} \sum_{k \in K} x_{0jk} \leq 2 \cdot |K| \tag{5}$$

$$\sum_{\substack{j \in V: \\ i<j}} x_{ijk} + \sum_{\substack{j \in V: \\ j<i}} x_{jik} = 2 \cdot z_{ik} \qquad \forall\, i \in V, k \in K \tag{6}$$

$$\sum_{i \in V \setminus \{0\}} s_{ip} \cdot u_{ipk} \leq q^{unit} \cdot y^{I}_{pk} \qquad \forall\, p \in P, k \in K \tag{7}$$

$$\sum_{p \in P} q^{unit} \cdot y^{I}_{pk} \leq Q \qquad \forall\, k \in K \tag{8}$$

$$\sum_{p \in P} y^{B}_{pk} \leq \hat{m} \qquad \forall\, k \in K \tag{9}$$

$$y^{I}_{pk} \leq \frac{Q}{q^{unit}} \cdot y^{B}_{pk} \qquad \forall\, p \in P, k \in K \tag{10}$$

$$\sum_{i \in S} \sum_{\substack{j \in S: \\ i<j}} x_{ijk} \leq |S| - 1 \qquad \forall\, k \in K, S \subseteq V \setminus \{0\}, |S| > 2 \tag{11}$$

$$u_{ipk} \in \{0,1\} \qquad \forall\, i \in V \setminus \{0\}, p \in P, k \in K \tag{12}$$

$$x_{0jk} \in \{0,1,2\} \qquad \forall\, j \in V \setminus \{0\}, k \in K \tag{13}$$

$$x_{ijk} \in \{0,1\} \qquad \forall\, i \in V \setminus \{0\}, j \in V \setminus \{0\}, i < j, k \in K \tag{14}$$

$$y^{B}_{pk} \in \{0,1\} \qquad \forall\, p \in P, k \in K \tag{15}$$

$$y^{I}_{pk} \in \left\{0,1,\dots,\left\lfloor \frac{Q}{q^{unit}} \right\rfloor\right\} \qquad \forall\, p \in P, k \in K \tag{16}$$

$$z_{ik} \in \{0,1\} \qquad \forall\, i \in V, k \in K. \tag{17}$$

The objective function (1) describes the total cost of all tours, which has to be minimized. Constraints (2) ensure that each supply is assigned to exactly one vehicle, whereas constraints (3) ensure that if a supply is assigned to a specific vehicle, also the respective location is visited by this vehicle. Constraints (4) guarantee that the depot is included on each tour, while constraint (5) limits the maximal number of vehicles to be used. Constraints (6) represent the node degree constraints. Constraints (7) define the capacity constraints of the compartments, whereas constraints (8) describe the capacity constraints of the vehicles. Constraints (9) limit the number of compartments in a vehicle to the maximal number of compartments. Constraints (10) define the relation between binary variables y^B_{pk} and integer variables y^I_{pk}. Constraints (11) are the subtour elimination constraints. Finally, constraints (12) to (17) define the variable domains.

The above formulation differs from the one presented by Henke et al. (2015) mainly with respect to the integer variables y^I_{pk} and the binary variables y^B_{pk}. Instead of using these variables, in the formulation by Henke et al. (2015) only binary variables y_{pkm} have been used. These variables indicate whether product type p is assigned to a compartment of vehicle k with compartment size m, where m indicated the selected size out of a set of potential compartment sizes. Accordingly, constraints (7), (8), (9), and (10) have been either modified or added. The changes resulted in significantly improved computing times, as will be shown in Section 5.5.

3 Literature Review

In recent years, several variants of the MCVRP have been discussed in the literature. Coelho and Laporte (2015) present a classification scheme, motivated by the context of fuel distribution, which distinguishes between split and unsplit compartments on the one hand and between split and unsplit tanks on the other hand. Split compartments indicate that the content of one compartment can be distributed to more than one customer location, whereas this is not the case with unsplit compartments. Similarly, split tanks indicate that a demand for the same product type of one customer location can be satisfied by more than one vehicle. In addition to this classification they propose models and a branch-and-cut algorithm which is able to solve all problem variants presented by the authors. Another branch-and-cut algorithm is proposed by Lahyani et al. (2015), who describe a problem in the context of olive oil collection. They consider fixed compartments sizes and a heterogeneous vehicle fleet.

El Fallahi et al. (2008) introduce a problem which arises in the context of animal food distribution. They consider fixed compartment sizes and an a priori fixed assignment of product types to compartments. In order to solve this problem, they present a memetic algorithm and a tabu search. A similar problem variant is considered by Muyldermans and Pang (2010) who introduce a guided local search. An application in the context of convenience store deliveries is described by Chajakis and Guignard (2003), for which a heuristic based on Lagrangean relaxation is proposed. In order to solve a similar problem, Reed et al. (2014) introduce an ant colony optimization algorithm, and Abdulkader et al. (2015) solve a distance-constrained variant of the problem by application of a hybrid ant colony optimization algorithm. Rabbani et al. (2016) consider a distance-constrained variant of the MCVRP with heterogeneous vehicles, multiple depots and mixed open and closed tours, for which they develop a genetic algorithm.

Another variant of the MCVRP, in which stochastic instead of deterministic demands are considered, is presented by Mendoza et al. (2010, 2011). They present several construction procedures and a memetic algorithm. Goodson (2015) proposes a simulated annealing algorithm for a similar problem. A multi-periodic variant of the stochastic problem with two product types is considered by Elbek and Wøhlk (2016). They introduce a heuristic which consists of a construction phase and a variable neighborhood search, which is implemented in a rolling planning horizon framework.

MCVRPs are frequently encountered in petrol replenishment. In these problems, compartment sizes are usually fixed in advance, but the assignment of product types to specific compartments is not. Since in most examples vehicles are not equipped with load meters, the content of each compartment can only be delivered to one customer. Such problems are described by Brown and Graves (1981) and Avella et al. (2004). A multi-periodic extension is considered by Cornillier et al. (2008) and Vidović et al. (2014).

Derigs et al. (2011) describe two variants of the MCVRP, namely problems with fixed compartment sizes on one hand and with flexible compartments sizes on the other. In both variants, the assignment of product types to compartments is a decision variable. They introduce a large neighborhood search which is able to solve all of their considered problem variants. A problem in the context of milk collection with fixed compartment sizes, a heterogeneous vehicle fleet, trailers, and without an a-priori assignment of product types to compartments is described by Caramia and Guerriero (2010). They present a two-part heuristic.

Another frequent application of MCVRPs occurs in maritime transportation (see for example Fagerholt and Christiansen 2000). Theoretical comparison between vehicle routing problems with one product type and multiple product types is presented by Archetti et al. (2016). They give theoretical and empirical findings between three variants of vehicle routing problems, namely the split delivery vehicle routing problem, the multi-compartment vehicle routing problem, and a combined split delivery and multi-compartment vehicle routing problem.

Overall, specific problems considered in the MCVRP context are quite heterogeneous. To the best of our knowledge, the only contribution which deals with discretely flexible compartment sizes and no a priori given assignment of product types to vehicles is the paper by Henke et al. (2015), who propose a variable neighborhood search to solve the MCVRP-DFCS. A slightly different problem with continuously flexible compartment sizes is considered by Koch et al. (2016). They propose a genetic algorithm to solve this variant. However, no specific exact algorithm has been developed in order to solve the MCVRP-DFCS. This paper aims at closing this gap.

4 Branch-and-Cut Algorithm

4.1 Overview

The formulation described in Section 2 has been solved with a branch-and-cut algorithm. The model formulation has been enhanced by adding several types of valid inequalities, capacity cuts and a specific branching scheme. Moreover, the subtour elimination constraints (11) are not included from the beginning but are instead added by application of a separation procedure. The components of the algorithm are described in detail in the following subsections.

4.2 Separation of Subtour-Elimination Constraints

As the number of subtour elimination constraints grows exponentially with the problem size, only for small instances all constraints can be generated and added to the model by a solver. A frequently-used option for overcoming this drawback consists of starting the solution process without any subtour-elimination constraints and then adding them dynamically during the solution process by means of a separation procedure. For the algorithm described in this paper, a simple procedure has been implemented: Each time an integer solution is found for a problem at a particular node, the solution is searched for subtours. If the solution possesses at least one subtour, the subtour with the smallest number of vertices and the respective set of vertices *S* is determined. Subsequently, the subtour-elimination constraint (11) for this set *S* is added to the pool of constraints and the problem represented by the respective node is solved again.

4.3 Valid Inequalities

To further strengthen the model formulation, several types of valid inequalities have been derived or adapted from the literature. On one hand, logical inequalities in the form of big-M-constraints have been developed which describe valid relations between two types of variables; on the other hand, symmetry breaking constraints have been adapted from the literature. All constraints have been tested with respect to how they contribute to decreasing the computing time. A subset of the most beneficial constraints has been implemented in the branch-and-cut algorithm.

For developing logical valid inequalities, all pairs of decision variables have been considered which have at least two indices in common. For a specific pair of variables, a big-M-constraint was derived if no constraint had already been included in the basic model which describes a relation between these variables. Table 1 indicates for each pair of variables whether a corresponding set of constraints has already been part of the basic model (respective inequalities given in italics), or whether a new set of valid inequality has been derived (respective inequalities underlined).

$\ldots \leq \ldots$	$\mathbf{u_{ipk}}$	$\mathbf{x_{ijk}}$	$\mathbf{y^B_{pk}}$	$\mathbf{y^I_{pk}}$	$\mathbf{z_{ik}}$
$\mathbf{u_{ipk}}$	-	<u>(18)</u>	<u>(19)</u>	*(7)*	*(3)*
$\mathbf{x_{ijk}}$	<u>(20)</u>	-	-	-	*(6)*
$\mathbf{y^B_{pk}}$	<u>(21)</u>	-	-	<u>(22)</u>	-
$\mathbf{y^I_{pk}}$	<u>(23)</u>	-	*(10)*	-	-
$\mathbf{z_{ik}}$	<u>(24)</u>	*(6)*	-	-	-

Table 1: Overview of relations between variables

The seven types of inequalities which have been newly derived are shown in the following explicitly. A detailed explanation for each type of constraint is not given as these constraints are only describing logical relationships between variables.

$$\sum_{p \in P} u_{ipk} \leq \frac{\hat{m}}{2} \cdot \left(\sum_{\substack{j \in V: \\ i<j}} x_{ijk} + \sum_{\substack{j \in V: \\ j<i}} x_{jik} \right) \qquad \forall\, i \in V \setminus \{0\}, k \in K \qquad (18)$$

$$\sum_{i \in V \setminus \{0\}} u_{ipk} \leq |V| \cdot y^B_{pk} \qquad \forall\, p \in P, k \in K \qquad (19)$$

$$\sum_{\substack{j \in V: \\ i<j}} x_{ijk} + \sum_{\substack{j \in V: \\ j<i}} x_{jik} \leq 2 \cdot \sum_{p \in P} u_{ipk} \qquad \forall\, i \in V \setminus \{0\}, k \in K \qquad (20)$$

$$y^B_{pk} \leq \sum_{i \in V \setminus \{0\}} u_{ipk} \qquad \forall\, p \in P, k \in K \qquad (21)$$

$$y^B_{pk} \leq y^I_{pk} \qquad \forall\, p \in P, k \in K \qquad (22)$$

$$y^I_{pk} \leq \frac{Q}{q^{unit}} \cdot \sum_{i \in V \setminus \{0\}} u_{ipk} \qquad \forall\, p \in P, k \in K \qquad (23)$$

$$z_{ik} \leq \sum_{p \in P} u_{ipk} \qquad \forall\, i \in V \setminus \{0\}, k \in K. \qquad (24)$$

In addition to the logical valid inequalities, also symmetry breaking constraints have been adapted from the literature. With respect to the MCVRP-DFCS, symmetry can occur between two solutions when all tours are identical but the assigned vehicles are different. Symmetry within a tour cannot occur as the undirected formulation has been used. In order to overcome the symmetry between solutions, the following three types of constraints have been adapted and tested:

$$z_{0,k+1} \leq z_{0k} \qquad \forall\, k \in K \setminus \{|K|\} \qquad (25)$$

$$\sum_{i \in V} \sum_{\substack{j \in V: \\ i<j}} c_{ij} \cdot x_{ij,k+1} \leq \sum_{i \in V} \sum_{\substack{j \in V: \\ i<j}} c_{ij} \cdot x_{ijk} \qquad \forall\, k \in K \setminus \{|K|\} \qquad (26)$$

$$\sum_{i \in V \setminus \{0\}} \sum_{p \in P} s_{ip} \cdot u_{ip,k+1} \leq \sum_{i \in V \setminus \{0\}} \sum_{p \in P} s_{ip} \cdot u_{ipk} \qquad \forall\, k \in K \setminus \{|K|\}. \qquad (27)$$

Constraints (25) ensure that a vehicle with index k+1 can only be used if the vehicle with index k is also used. Constraints (26) ensure that tours are ordered in such a way that a tour with a higher index cannot have a total cost higher than a tour with a lower index. Similarly, constraints (27) order the tours with respect to non-decreasing total supplies.

A set of preliminary experiments allowed us to determine that a combination of the logical valid inequalities (19), (22), (23) leads to the highest reduction of the computing time. Regarding the symmetry breaking constraints, constraints (26) have shown to be most beneficial. Therefore, these four types of inequalities have been added to the basic model.

4.4 Capacity Cuts

As additional valid inequalities, also an adapted version of the well-known capacity cuts has been added, which were introduced by Cornuejols and Harche (1993) for the CVRP. Equation (28) shows the original form of these cuts. The basic idea is that the number of selected edges between a subset of nodes *S* which does not include the depot and the set of all remaining nodes *V\S* must be larger than or equal to a theoretical minimum on the number of vehicles which are necessary for satisfying the demands of all nodes in subset *S*:

$$\sum_{i \in S} \sum_{\substack{j \in V \setminus S: \\ i<j}} \sum_{k \in K} x_{ijk} + \sum_{i \in S} \sum_{\substack{j \in V \setminus S: \\ j<i}} \sum_{k \in K} x_{jik} \geq 2 \cdot \left\lceil \frac{\sum_{i \in S \setminus \{0\}} d_i}{Q} \right\rceil \qquad \forall\, S \subseteq V \setminus \{0\},\, |S| > 1. \tag{28}$$

The adapted capacity cuts for the MCVRP-DFCS are shown in equations (29). The left-hand side, which is similar to the original capacity cuts for the CVRP, must be greater than or equal to the maximum of two alternative lower bounds, LB_1 and LB_2:

$$\sum_{i \in S} \sum_{\substack{j \in V \setminus S: \\ i<j}} \sum_{k \in K} x_{ijk} + \sum_{i \in S} \sum_{\substack{j \in V \setminus S: \\ j<i}} \sum_{k \in K} x_{jik} \geq \max\{LB_1, LB_2\} \qquad \forall\, S \subseteq V \setminus \{0\},\, |S| > 1. \tag{29}$$

LB_1 (see equation (30)) is based on the original idea of the capacity cuts but, additionally, takes into account that compartment capacities can only be multiples of the basic compartment unit size:

$$LB_1 = 2 \cdot \left\lceil \frac{\sum_{p \in P} \left\lceil \frac{\sum_{i \in S} s_{ip}}{q^{unit}} \right\rceil \cdot q^{unit}}{Q} \right\rceil. \tag{30}$$

LB_2 (see equation (31)), in contrast to LB_1, focuses on the maximum number of compartments. The numerator in the fraction of (31) corresponds to a lower bound on the number of compartments which are necessary in order to collect all supplies, whereas the denominator indicates the maximum number of compartments into which the vehicle capacity can be separated. In the numerator, two different cases are distinguished in which the use of a compartment becomes necessary. Parameter $\grave{s}_p$ indicates whether there is at least one supply for a certain product type in subset *S* (case 1). Moreover, parameter $\tilde{s}_p$ indicates whether it might be necessary to use more than only one compartment for a certain product type (case 2) and, if so, the parameter takes the value of the additional number of compartments needed.

At least more than one compartment for a single product type is needed, when the sum of supplies for this product type in S is greater than the vehicle capacity:

$$LB_2 = 2 \cdot \left\lceil \frac{\sum_{p \in P} (\grave{s}_p + \tilde{s}_p)}{\widehat{m}} \right\rceil \qquad \grave{s}_p = \begin{cases} 1, \text{ if} \sum\limits_{i \in S} s_{ip} > 0, \\ 0, \text{ else.} \end{cases} \qquad \tilde{s}_p = \left\lfloor \frac{\sum_{i \in S} s_{ip}}{Q} \right\rfloor. \tag{31}$$

Let, for example, a vehicle with two compartments and supplies for four different product types (A, B, C, and D) in subset S be given. Then, according to case 1, at least $\lceil (1+1+1+1)/2 \rceil = 2$ vehicles are necessary to collect all supplies. If, moreover, the total supply of product type A would amount to 1.2Q, then at least two compartments in two different vehicles are necessary in order to collect all supplies of product type A. Thus, at least $\lceil (4+1)/2 \rceil = 3$ vehicles would be needed to collect all supplies.

As the number of capacity cuts increases exponentially with the number of nodes in the problem, in order to separate these cuts, we use the shrinking heuristic introduced by Ralphs et al. (2003).

4.5 Branching Scheme

Finally, the sequence according to which variables are selected for branching has been further specified. Having selected any node for branching, the y^B_{pk}-variables are considered first. Only if all y^B_{pk}-variables are already integral, branching according to z_{ik}-variables is considered, then according to x_{ijk}-variables and finally according to u_{ipk}- and y^I_{pk}-variables with same priority. This sequence has been determined in preliminary tests; it is comprehensible, too, since the overall impact of a decision variable on the composition of a solution decreases according to this sequence. However, the stated sequence is changed slightly if instances are considered in which the number of compartments is not smaller than the number of product types. In such a case, the highest priority is given to the z_{ik}-variables, then to the x_{ijk}-variables, and finally y^B_{pk}-variables, u_{ipk}- and y^I_{pk}-variables with same priority.

5 Computational Experiments

5.1 Overview

In order to evaluate the performance of the branch-and-cut algorithm proposed for the MCVRP-DFCS, two sets of experiments have been performed. In the first set, related to previously published instances with 10 customer locations, the performance of the branch-and-cut algorithm is compared to the performance of the exact solution approach presented in Henke et al. (2015). In the second set of experiments, the performance of the algorithm on larger problem instances is analyzed in greater detail. For this purpose, new instances with up to 50 locations have been generated randomly. Furthermore, the contribution of specific algorithmic features to the performance of the branch-and-cut algorithm are studied on a subset of the newly generated instances. A third set of experiments has been introduced in order to gain additional managerial insights. In these experiments the costs have been investigated which result from the use of discretely flexible compartments instead of continuously flexible compartments.

The algorithm has been implemented on a 3.2 gigahertz and 8 gigabytes RAM computer using C++ and the interface with CPLEX 12.6. For the separation procedure, a lazy-cut-callback procedure has been used, whereas the capacity cuts are added within a user-cut-callback procedure.

5.2 Problem Instances

For the first set of experiments, the instances previously introduced by Henke et al. (2015) (called "small instances" here) have been used. These instances are characterized by 10 customer locations and differ with respect to three parameters: the number of product types $|P|$, the maximal number of compartments $\widehat{m}$, and a supply parameter $\bar{\mathbf{s}}$. The latter indicates whether the total number of positive supplies is small (1), medium (2), or large (3). For the number of product types $|P|$, 3, 6, and 9 product types, respectively, have been assumed. For the maximal number of compartments $\widehat{m}$, 2 or 3 compartments have been considered for instances with three product types, 2, 4 or 6 compartments for instances with 6 product types; and 2, 4, 7 or 9 compartments for instances with 9 product types. This gives rise to 27 problem classes, for which Henke et al. (2015) generated 50 instances each. Thus, in total, this problem set includes 1,350 instances.

For the second set of experiments, 675 larger instances with up to 50 locations were newly generated. The instances have been obtained by means of a slight modification of the instance generator described by Henke et al. (2015). The modification concerned the vehicle capacity and the average number of locations assigned to a vehicle. While Henke et al. (2015) generated instances in which all vehicle capacities were fixed to the same value and the supplies were generated in such a way that the average number of locations per vehicle remained constant, we decided to consider the number of locations as a problem parameter to be investigated and, thus, introduced a flexible vehicle capacity dependent on the number of locations. As a result, the number of vehicles in the obtained instances remains relatively constant. Furthermore, the parameter specifications on the number of product types and the maximal number of compartments have been changed. For the newly generated instances either 3 or 4 product types were assumed. In case of instances with three product types, 2 or 3 compartments have been considered, while in case of four product types the number of compartments has been set to 2, 3 or 4. For the experiments with respect to the evaluation of the algorithmic components, a subset of 50 instances was selected randomly from the newly generated instances in such a way that for each considered number of locations, at least two instances were included in the subset. As we expected the computing times to increase substantially for these experiments, only instances were considered for which the computing time of the branch-and-cut algorithm did not exceed ten minutes. The same set of instances has been used for the third set of experiments.

5.3 Experiments with Instances from the Literature

Regarding the small instances from Henke et al. (2015), the branch-and-cut algorithm was able to solve all instances to optimality. Table 2 shows the detailed results clustered according to the number of product types, the maximal number of compartments, and the supply parameter. The table also depicts the average total cost for each cluster (avg.tc), the average computing time for the method of Henke et al. (2015) (cpu.old), the average computing time of the proposed branch-and-cut algorithm (cpu.new), and the average improvement in computing time (cpu.impr). It can be observed that the branch-and-cut algorithm provides a significant improvement of the computing times, which, on average, decreased by 87.19% from 431.31 seconds to 43.02 seconds across all instances.

parameters			**avg.tc**	**cpu.old**	**cpu.new**	**cpu.impr**
$\lvert \boldsymbol{P} \rvert$	$\hat{\mathbf{m}}$	$\bar{\mathbf{s}}$				
3	2	1 (small)	366.90	5.43	0.27	95.10%
		2 (medium)	465.35	11.43	0.32	97.24%
		3 (large)	596.16	101.33	0.35	99.65%
	3	1 (small)	349.08	2.60	0.17	93.58%
		2 (medium)	337.15	2.66	0.19	92.77%
		3 (large)	336.64	5.00	0.19	96.15%
6	2	1 (small)	549.98	5.91	0.45	92.39%
		2 (medium)	767.48	4.83	0.65	86.62%
		3 (large)	919.78	1.81	0.88	51.67%
	4	1 (small)	366.98	5.04	0.72	85.63%
		2 (medium)	492.11	27.79	1.46	94.75%
		3 (large)	583.82	326.15	4.57	98.60%
	6	1 (small)	351.98	4.05	0.32	92.10%
		2 (medium)	357.98	5.94	0.49	91.80%
		3 (large)	357.42	8.96	0.57	93.69%
9	2	1 (small)	724.17	502.44	37.83	92.47%
		2 (medium)	1,114.25	1,921.59	127.39	93.37%
		3 (large)	1,419.09	5,400.85	646.32	88.03%
	4	1 (small)	463.05	12.87	2.44	81.01%
		2 (medium)	647.57	267.69	11.59	95.67%
		3 (large)	815.27	1,324.58	20.15	98.48%
	7	1 (small)	360.16	5.25	5.23	0.41%
		2 (medium)	456.33	93.21	8.57	90.81%
		3 (large)	564.50	1,550.47	287.27	81.47%
	9	1 (small)	354.98	3.94	0.58	85.28%
		2 (medium)	392.48	10.53	1.05	90.08%
		3 (large)	383.36	32.94	1.56	95.27%
minimum			336.64	1.81	0.17	0.41%
average			551.63	431.31	43.02	87.19%
maximum			1,419.09	5,400.85	646.32	99.65%

Table 2: Results for the instances from the literature

5.4 Experiments with New Instances

5.4.1 Evaluation of the Algorithm on Larger Instances

For the experiments with the newly generated instances, a time limit of two hours has been set for each instance. The results for the 675 instances have been clustered according to the four problem parameters, i.e. the number of locations (n), the number of product types ($|P|$), the maximum number of compartments ($\hat{m}$), and the supply parameter ($\bar{s}$), in Tables 3 to 6, respectively. In each table and for each cluster the number of instances in the cluster (#inst), the average total cost (avg.tc), the number of optimally solved instances in the cluster (#opt), the relative number of optimally solved instances per cluster (rel.opt), the average gap (avg.gap), the maximal gap (max.gap), as well as the minimal (min.cpu), average (avg.cpu), and maximal computing times (max.cpu) are given.

Table 3 shows the results clustered according to the number of locations. As expected, the performance of the algorithm decreases with an increasing number of locations. For all instances with up to 25 locations the algorithm was able to determine optimal solutions. For the remaining instances the relative number of optimally solved instances decreases from 97.3% for instances with 30 locations down to 66.7% for instances with 50 locations, i.e. even for the largest instances the algorithm was able to find optimal solutions for two out of three instances. Similar changes can be observed with respect to the gaps and computing times. The average gap ranges from 0.04% for instances with 30 locations to 1.71% for instances with 50 locations, and the average computing times from 0.51 seconds for instances with 10 locations to 2,883.37 seconds for instances with 50 locations.

n	#inst	avg.tc	#opt	rel.opt	avg.gap	max.gap	min.cpu	avg.cpu	max.cpu
10	75	447.62	75	100.0%	0.00%	0.00%	0.06	0.51	5.82
15	75	518.57	75	100.0%	0.00%	0.00%	0.06	1.47	18.25
20	75	566.24	75	100.0%	0.00%	0.00%	0.08	5.82	56.87
25	75	626.30	75	100.0%	0.00%	0.00%	0.17	95.69	4,207.43
30	75	663.04	73	97.3%	0.04%	1.90%	0.23	312.57	7,201.13
35	75	710.30	71	94.7%	0.08%	2.13%	0.30	793.22	7,207.21
40	75	750.23	61	81.3%	0.71%	8.16%	0.62	1,609.24	7,202.49
45	75	781.40	52	69.3%	1.38%	11.20%	1.25	2,625.75	7,202.75
50	75	812.14	50	66.7%	1.71%	16.00%	0.66	2,883.37	7,219.71

Table 3: Results for the generated instances clustered according to the number of locations

Table 4 summarizes the results for all instances clustered according to the number of product types. Although the differences between the two clusters are not very large, it can be observed that the algorithm performs better on instances with a smaller number of product types. The relative number of optimally solved instances decreases from 92.6% to 88.1%, the average gap increases from 0.34% to 0.50%, and the average computing time increases from 694.39 seconds to 1,079.23 seconds.

$\|P\|$	#inst	avg.tc	#opt	rel.opt	avg.gap	max.gap	min.cpu	avg.cpu	max.cpu
3	270	628.24	250	92.6%	0.34%	9.35%	0.06	694.39	7,202.51
4	405	669.29	357	88.1%	0.50%	16.00%	0.09	1,079.23	7,219.71

Table 4: Results for the generated instances clustered according to the number of product types

Table 5 demonstrates the effect of the maximum number of compartments on the performance of the algorithm. Instead of presenting the results clustered according to this parameter only, also the number of product types has been taken into account. We do so because a similar maximum number of compartments might imply different consequences on the problem structure when the number of product types is different. For example, for instances with three product types and instances with four product types, it can be observed that the performance of the algorithm improves with an increasing maximum number of compartments, although the maximum number of compartments does not have an impact on the problem size. An explanation for this is that the decision of assigning product types to vehicles becomes more difficult when fewer compartments are available. Similar findings have also been reported by Henke et al. (2015). For the case with three product types, the relative number of optimally solved instances increases from 82.5% with two compartments to 100.0% with three compartments. For the case with four product types, only 82.2% of the instances with two compartments were solved to optimality, whereas 97.8% of the instances with four compartments were solved to optimality. The findings with respect to gaps and computing times show similar trends. When clusters with the same maximum number of compartments, but different product types are compared, the performance of the algorithm decreases slightly when the number of product types increases, which further underlines the findings from Table 4.

$\lvert P \rvert$	$\hat{m}$	#inst	avg.tc	#opt	rel.opt	avg.gap	max.gap	min.cpu	avg.cpu	max.cpu
3	2	135	740.14	115	85.2%	0.68%	9.35%	0.06	1,221.94	7,202.51
	3	135	516.34	135	100.0%	0.00%	0.00%	0.07	166.83	3,496.57
4	2	135	808.37	111	82.2%	0.80%	16.00%	0.22	1,499.65	7,219.71
	3	135	674.38	114	84.4%	0.65%	11.33%	0.10	1,434.20	7,209.30
	4	135	525.12	132	97.8%	0.05%	2.59%	0.09	303.84	7,202.20

Table 5: Results for the generated instances clustered according to the number of product types and the maximal number of compartments

Finally, Table 6 shows the results clustered according to the supply parameter. Here, no clear trend can be observed. The performance of the algorithm seems to be best when the number of supplies is large. However, in general, the number of supplies has little effect on the performance of the algorithm.

$\bar{s}$	#inst	avg.tc	#opt	rel.opt	avg.gap	max.gap	min.cpu	avg.cpu	max.cpu
1	225	561.16	202	89.8%	0.56%	16.00%	0.09	855.36	7,219.71
2	225	654.94	200	88.9%	0.30%	9.35%	0.14	1,085.94	7,202.75
3	225	742.50	205	91.1%	0.45%	9.22%	0.06	834.58	7,202.50

Table 6: Results for the generated instances clustered according to the number of supplies

5.4.2 Evaluation of the Components of the Algorithm

In order to investigate the different components of the algorithm further, we conducted experiments on a subset of 50 of the newly generated instances. In the first part of these experiments, the impact of the change in the model formulation (see Section 2) has been analyzed. For this purpose, the 50 instances have been solved by implementing the formulation of Henke et al. (2015) and the modified formulation proposed in this paper without any added components, i.e. without valid inequalities, capacity cuts, or variable priorities. Table 7 shows the corresponding results, clustered according to the number of locations. For each cluster, the table indicates the number of instances (#inst), the number of instances which have been solved to optimality by the formulation of Henke et al. (2015) (#opt.old) and by the formulation introduced in Section 2 (#opt.new), the average gaps achieved by each implemented model (gap.old, gap.new) and the gap improvement (gap.impr), as well as the average computing times for each formulation (cpu.old, cpu.new) and the respective reduction of computing time (cpu.impr). The new formulation outperformed the old formulation in all clusters. Although its implementation was only able so solve three additional instances to optimality, the reduction of the average gaps from 5.68% to 2.79% and the reduction of the average computing times from 4,014.74 seconds to 2,904.60 seconds demonstrate the superiority of the changes in the model formulation.

n	#inst	#opt.old	#opt.new	gap.old	gap.new	gap.impr	cpu.old	cpu.new	cpu.impr
10	2	2	2	0.01%	0.01%	1.35%	2541.77	11.33	99.55%
15	5	5	5	0.01%	0.01%	28.03%	512.92	10.19	98.01%
20	7	6	6	1.93%	0.66%	65.66%	1931.11	1083.13	43.91%
25	5	4	4	2.63%	0.41%	84.39%	3179.51	1519.00	52.23%
30	5	2	3	6.12%	1.09%	82.20%	4452.54	3034.38	31.85%
35	7	0	0	15.75%	8.05%	48.86%	7213.26	7206.70	0.09%
40	8	3	4	8.10%	4.56%	43.74%	5374.59	3847.53	28.41%
45	6	2	2	8.79%	5.02%	42.87%	4923.36	4858.49	1.32%
50	5	1	2	7.75%	5.29%	31.81%	6003.58	4570.65	23.87%
sum		25	28						
average				5.68%	2.79%	47.66%	4014.74	2904.60	42.14%

Table 7: Comparison of model formulations

In each of the experiments summarized in Table 8, exactly one algorithmic component has been eliminated from the algorithm, e.g. valid inequalities were not included or the specified branching scheme has not been defined. The first column of Table 8 shows the component which has been neglected. These components are the symmetry breaking constraints, the three types of logical valid inequalities, the capacity cuts, and the definition of variable priorities. In the second column, the average computing times (avg.cpu) are given. In column 3, the relative number of optimally solved instances (rel.opt) is listed and, in column 4, the average gap (avg.gap) is given. As indicated in the last row, the algorithm with all components was able to solve all 50 instances to optimality and, therefore, the relative number of optimally solved instances amounts to 100% and the corresponding average gap is 0.00%. It can be observed that the elimination of any of these six components leads to an increase in the average computing time and a decrease in the overall algorithmic performance. Interestingly, the contribution to the algorithmic performance is very different for each component. Elimination of the capacity cuts (elimination of the variable priorities) increased the average computing times from 81.48 seconds to 1,851.14 seconds (543.80 seconds), while the effect resulting from an elimination of the remaining components is much smaller. With respect to the number of optimally solved instances, similar findings can be identified. Only 80% of the instances were solved to optimality when the capacity cuts were eliminated, 96% when the variable priorities were neglected, and 98% when valid inequalities (23) were removed from the formulation. For the remaining three components, no change occurred. Finally, the average gaps show similar results. Without capacity cuts an average gap of 1.45% occurred, 0.12% without variable priorities, 0.02% without valid inequalities (23), and 0.00% for the remaining components. Still these components lead to an increase in the performance of the algorithm and have, therefore, been included in its final version.

Relaxed component	avg.cpu	rel.opt	avg.gap
all components	3,273.07	56.00%	3.23%
Capacity cuts	1,851.14	80.0%	1.45%
Variable priorities	543.80	96.0%	0.12%
Valid inequalities (23)	205.44	98.0%	0.02%
Valid inequalities (19)	125.14	100.0%	0.00%
Symmetry breaking constraints (26)	90.37	100.0%	0.00%
Valid inequalities (22)	68.19	100.0%	0.00%
no relaxation	81.48	100.00%	0.00%

Table 8: Results for the design experiments

5.5 Cost of Discretization

In order to gain additional managerial insights related to the problem specific property of compartment discretization, the branch-and-cut algorithm has also been used for an experiment in which the total cost with continuously flexible compartment sizes were compared to the total cost of discretely flexible compartment sizes. To obtain optimal solutions for the case of continuously flexible compartment sizes, the only necessary change in the branch-and-cut algorithm was to set the basic unit compartment size equal to 1.

Table 9 displays the findings of this experiment clustered according to the number of locations. In the second column, the average total cost for the case of continuously flexible compartments (avg.tc) is indicated, and in the third column the average reduction of the total cost compared to the discretely flexible case (red.tc) is shown.

n	avg.tc	red.tc
10	458.31	5.89%
15	485.13	4.23%
20	610.41	0.59%
25	628.06	3.41%
30	666.72	1.97%
35	627.07	2.98%
40	672.97	1.10%
45	717.70	0.52%
50	727.08	0.81%

Table 9: Cost of discretization

Overall the total cost can be reduced by 2.39% when discretely flexible compartments are substituted by continuously flexible compartments. It can be observed that the potential for cost savings tends to decrease with the number of locations. Whereas 5.89% of total cost can be saved on instances with 10 locations, only 0.81% can be saved on instances with 50 locations.

Further analyses have shown that the saving potential is higher with an increasing number of product types (1.25% for three product types, 2.42% for four product types) and an increasing maximum number of compartments (0.41% for two compartments, 2.43% for three compartments, and 3.97% for four compartments). Again, for the supply parameter no general trend could be observed.

6 Summary and Outlook

In this paper, a multi-compartment vehicle routing problem with discretely flexible compartment sizes (MCVRP-DFCS) has been investigated and a branch-and-cut algorithm has been proposed. The presented algorithm has been tested in extensive numerical experiments based on 1,350 instances from the literature and 675 newly generated instances. The results show that the proposed method outperforms the existing method by 87.19% on average with respect to the computing time. The algorithm is able to solve all newly generated instances with up to 25 locations and 2/3 of instances with 50 locations to optimality within two hours. Even for instances with 50 locations the average gap amounted to 1.71% only. Additional experiments show that savings of 2.39% can be achieved if vehicles with continuous compartment sizes are used instead of vehicles with discrete compartment sizes.

Further research directions related to the MCVRP-DFCS are, amongst others, to consider a multi-periodic context and stochastic supplies instead of deterministic ones. Both aspects can be found in the real-world application of glass waste collection but have not yet been investigated in this context.

References

Abdulkader, M.M.S., Gajpal, Y., El Mekkawy, T.Y. (2015): Hybridized ant colony algorithm for the multi compartment vehicle routing problem. Applied Soft Computing 37, 196-203.

Archetti, C., Campbell, A., Speranza, M.G. (2016): Multi-commodity vs. single-commodity routing. Transportation Science 50, 461-472.

Avella, P., Boccia, M., Sforza, A. (2004): Solving a fuel delivery problem by heuristic and exact approaches. European Journal of Operational Research 152, 170-179.Brown, G.G., Graves, G.W. (1981): Real-time dispatch of petroleum tank trucks. Management Science 27, 19-32.

Caramia, M., Guerriero, F. (2010): A milk collection problem with incompatibility constraints. Interfaces 40, 130-143.

Chajakis, E.D., Guignard, M. (2003): Scheduling deliveries in vehicles with multiple compartments. Journal of Global Optimization 26, 43-78.

Coelho, L.C., Laporte, G. (2015): Classification, models and exact algorithms for multi-compartment delivery problems. European Journal of Operational Research 242, 854-864.

Cornillier, F., Boctor, F.F., Laporte, G., Renaud, J. (2008): A heuristic for the multi-period petrol station replenishment problem. European Journal of Operational Research 191, 295-305.

Cornuejols, G., Harche, F. (1993): Polyhedral study of the capacitated vehicle routing problem. Mathematical Programming 60, 21-52.

Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., Vogel, U. (2011): Vehicle routing with compartments: applications, modelling and heuristics. OR Spectrum 33, 885-914.

El Fallahi, A., Prins, C., Wolfer Calvo, R. (2008): A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Computers & Operations Research 35, 1725-1741.

Elbek, M., Wøhlk, S. (2016): A variable neighborhood search for the multi-period collection of recyclable materials. European Journal of Operational Research 249, 540-550.

Fagerholt, K., Christiansen, M. (2000): A combined ship scheduling and allocation problem. Journal of the Operational Research Society 51, 834-842.

Golden, B.L., Raghavan, S., Wasil, E.A. (2008): The vehicle routing problem: latest advances and new challenges. New York: Springer.

Goodson, J.C. (2015): A priori policy evaluation and cyclic-order-based simulated annealing for the multi-compartment vehicle routing problem with stochastic demands. European Journal of Operational Research 241, 361-369.

Henke, T., Speranza, M.G., Wäscher, G. (2015): The multi-compartment vehicle routing problem with flexible compartment sizes. European Journal of Operational Research 246, 730-746.

Koch, H., Henke, T., Wäscher, G. (2016): A genetic algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. Working Paper No. 04/2016, Fakultät für Wirtschaftswissenschaft, Otto-von-Guericke Universität Magdeburg.

Lahyani, R., Coelho, L.C., Khemakhem, M., Laporte, G., Semet, F. (2015): A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. Omega 51, 1-10.

Laporte, G. (2009): Fifty years of vehicle routing. Transportation Science 43, 408-416.

Mendoza, J.E., Castanier, B., Guéret, C, Medaglia, A.L., Velasco, N. (2010): A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. Computers & Operations Research 37, 1886-1898.

Mendoza, J.E., Castanier, B., Guéret, C., Medaglia, A.L., Velasco, N. (2011): Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. Transportation Science 45, 346-363.

Muyldermans, L., Pang, G. (2010): On the benefits of co-collection: experiments with a multi-compartment vehicle routing problem. European Journal of Operational Research 206, 93-103.

Rabbani, M., Farrokhi-asl, H., Rafiei, H. (2016): A hybrid genetic algorithm for waste collection problem by heterogeneous fleet of vehicles with multiple separated compartments. Journal of Intelligent & Fuzzy Systems 30, 1817-1830.

Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter, L.E. (2003): On the capacitated vehicle routing problem. Mathematical Programming, Series B 94, 343-359.

Reed, M., Yiannakou, A., Evering, R. (2014): An ant colony algorithm for the multi-compartment vehicle routing problem. Applied Soft Computing 15, 169-176.

Toth, P., Vigo, D. (2014): Vehicle routing: Problems, methods, and applications (2nd ed.). Philadelphia: Society for Industrial and Applied Mathematics.

Vidović, M., Popović, D., Ratković, B. (2014): Mixed integer and heuristics model for the inventory routing problem in fuel delivery. International Journal of Production Economics 147, 593-604.

V

A Genetic Algorithm for the Multi-Compartment Vehicle Routing Problem with Flexible Compartment Sizes

A Genetic Algorithm for the Multi-Compartment Vehicle Routing Problem with Flexible Compartment Sizes

Henriette Koch
Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany
henriette.koch@ovgu.de

Tino Henke
Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany
tino.henke@ovgu.de

Gerhard Wäscher
Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany
School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, 100044 Beijing, China
gerhard.waescher@ovgu.de

Abstract

In this paper, a genetic algorithm for the multi-compartment vehicle routing problem with continuously flexible compartment sizes is proposed. In this problem, supplies of several product types have to be collected from customer locations and transported to a depot at minimal cost. In order to avoid mixing of different product types which are transported in the same vehicle, the vehicle's capacity can be separated into a limited number of compartments. The size of each compartment can be selected arbitrarily within the limits of the vehicle's capacity, and in each compartment one or several supplies of the same product type can be transported.
For solving this problem, a genetic algorithm is presented. The performance of the proposed algorithm is evaluated by means of extensive numerical experiments. Furthermore, the economic benefits of using continuously flexible compartments are investigated.

Keywords: vehicle routing, multiple compartments, genetic algorithm, heuristics

1 Introduction

In the classic capacitated vehicle routing problem (CVRP), goods have to be transported from a central depot to several customers. Each customer requires to be supplied with a certain quantity of goods and the available vehicles have limited capacities (see e.g. Dantzig and Ramser 1959, or Toth and Vigo 2014). In this paper, a variant of the CVRP is considered in which different product types have to be collected from customer locations and kept separated during transportation. For this purpose, the loading space of the collection vehicle can be divided into different compartments, where in each compartment a single product type can be transported.

In general, this problem can be classified as a multi-compartment vehicle routing problem (MCVRP). Problems of this kind arise when products which require different transportation conditions (e.g. food of different levels of refrigeration, Derigs et al. 2011), bulk (Fagerholt and Christiansen 2000) or liquid products, like petroleum (Brown and Graves 1981), are considered. The collection of glass waste of different colours (Henke et al. 2015) constitutes another practical application. Previous research results suggest that substantial cost savings can be gained when vehicles with multiple compartments are used in such contexts compared to vehicles without compartments (Henke et al. 2015).

In most previously discussed MCVRP variants, compartment sizes are fixed and unchangeable. In contrast to this, a MCVRP with flexible compartment sizes is considered in this paper, i.e. the compartment sizes for each product type can be adjusted arbitrarily. Furthermore, the maximal number of compartments that can be used in one vehicle can be equal to the number of product types, but it can also be smaller. While compartment sizes in Henke et al. (2015) can only be selected from a set of potential compartment sizes (discrete flexibility) in their variant, we consider compartment sizes which can be selected arbitrarily (continuous flexibility). In the following, the regarded problem will be referred to as the multi-compartment vehicle routing problem with continuously flexible compartment sizes (MCVRP-CFCS). Apart from the typical CVRP decisions about the composition of the tours, i.e. which customers are visited by each vehicle and in which sequence, it needs to be decided for each vehicle which product types are transported and which compartment sizes are chosen.

As only small problem instances of the MCVRP-CFCS can be solved to optimality by exact solution approaches, heuristic solution approaches are more suitable for practically relevant problem sizes. In this paper, a new solution approach for the MCVRP-CFCS, namely a genetic algorithm, is presented and evaluated by means of on extensive numerical experiments. Furthermore, managerial insights into the benefits from using vehicles with continuously flexible compartments compared to vehicles with discretely flexible compartments are given.

The remainder of this paper is organized as follows: In Section 2, a formal definition and a mathematical model for the MCVRP-CFCS are presented. Section 3 gives an overview of the relevant literature about the MCVRP. In Section 4, the proposed genetic algorithm is introduced and explained. The algorithm was tested by means of extensive numerical experiments. The experimental design and the corresponding results will be presented in Section 5. Finally, the main findings are summarized in Section 6.

2 Problem Description and Formulation

The MCVRP-CFCS can be stated as follows (Henke et al. 2015): Let $G = (V, E)$ be an undirected, weighted graph where the vertex set $V = \{0, 1, \ldots, n\}$ represents the depot ($\{0\}$) and n customer locations, and $E = \{(i,j)\colon i, j \in V, i < j\}$ represents the set of edges which can be travelled between the locations. A non-negative cost c_{ij}, $(i, j) \in E$ is assigned to each edge. Furthermore, let P be a set of product types and $s_{ip} \geq 0$ the supply at location i of product type p ($i \in V \setminus \{0\}$, $p \in P$). These supplies have to be transported from the customer locations to the depot in separate product type-specific compartments. For this purpose, a set of homogenous vehicles K is available. Each of these vehicles has the same capacity Q which can be divided into a (limited) number of compartments $\hat{m} \leq |P|$. The size of a compartment can be selected arbitrarily between 0 and Q provided that the sum of the sizes of all compartments in a vehicle must not exceed its capacity. Since a vehicle might not be able to transport all product types at the same time, a customer location can be visited by more than one vehicle. However, an individual supply of a product type at a certain location must not be split and assigned onto several vehicles.

In order to solve this problem, several partial decisions have to be made simultaneously: It has to be determined which product types are to be delivered by each vehicle. Moreover, each supply has to be assigned to one tour which also contains the decision about the assignment of customer locations to tours and about the compartment sizes, i.e. the compartment size for product type p in vehicle k results from the sum of all supplies of product type p that are assigned to vehicle k. Finally, for each tour the visiting sequence of the locations assigned to the tour has to be determined. The objective of the problem is to minimize the total cost of travelling.

In order to formulate the mathematical model for the MCVRP-CFCS, the following decision variables are introduced:

$$u_{ipk} = \begin{cases} 1, \text{ if supply of product type p at location i is collected by vehicle k,} \\ 0, \text{ otherwise,} \end{cases} \qquad i \in V \setminus \{0\}, p \in P, k \in K;$$

$$x_{ijk} = \begin{cases} 2, \text{ if } i = 0 \text{ and edge (i,j) is used twice by vehicle k,} \\ 1, \text{ if edge (i,j) is used once by vehicle k,} \\ 0, \text{ otherwise,} \end{cases} \qquad i,j \in V\colon i < j, k \in K;$$

$$y_{pk} = \begin{cases} 1, \text{ if a compartment for product type p is used in vehicle k,} \\ 0, \text{ otherwise,} \end{cases} \qquad p \in P, k \in K;$$

$$z_{ik} = \begin{cases} 1, \text{ if location i is visited by vehicle k,} \\ 0, \text{ otherwise,} \end{cases} \qquad i \in V, k \in K.$$

The model can be stated as follows:

$$\min \; z(S) = \sum_{(i,j)\in E} \sum_{k\in K} c_{ij} \cdot x_{ijk} \tag{1}$$

$$\sum_{k\in K} u_{ipk} = 1 \qquad \forall\, i \in V\backslash\{0\},\, p \in P\!: s_{ip}>0 \tag{2}$$

$$u_{ipk} \le z_{ik} \qquad \forall\, i \in V\backslash\{0\},\, p \in P,\, k \in K \tag{3}$$

$$u_{ipk} \le y_{pk} \qquad \forall\, i \in V\backslash\{0\},\, p \in P,\, k \in K \tag{4}$$

$$z_{ik} \le z_{0k} \qquad \forall\, i \in V\backslash\{0\},\, k \in K \tag{5}$$

$$\sum_{\substack{j\in V:\\ j>0}} \sum_{k\in K} x_{0jk} \le 2 \cdot |K| \tag{6}$$

$$\sum_{\substack{j\in V:\\ i<j}} x_{ijk} + \sum_{\substack{j\in V:\\ j<i}} x_{jik} = 2 \cdot z_{ik} \qquad \forall\, i \in \mathrm{V},\, k \in K \tag{7}$$

$$\sum_{p\in P} y_{pk} \le \hat{m} \qquad \forall\, k \in K \tag{8}$$

$$\sum_{p\in P} \sum_{i\in V\backslash\{0\}} s_{ip} \cdot u_{ipk} \le Q \qquad \forall\, k \in K \tag{9}$$

$$\sum_{i\in V'} \sum_{\substack{j\in V':\\ i<j}} x_{ijk} \le |V'|\text{-}1 \qquad \forall\, k \in K,\, V' \subseteq V\backslash\{0\}\!:|V'|>2 \tag{10}$$

$$u_{ipk} \in \{0,1\} \qquad \forall\, i \in V\backslash\{0\},\, p \in P,\, k \in K \tag{11}$$

$$x_{0jk} \in \{0,1,2\} \qquad \forall\, j \in V\backslash\{0\},\, k \in K \tag{12}$$

$$x_{ijk} \in \{0,1\} \qquad \forall\, i \in V\backslash\{0\}, j \in V\backslash\{0\}\!: i<j,\, k \in K \tag{13}$$

$$y_{pk} \in \{0,1\} \qquad \forall\, p \in P,\, k \in K \tag{14}$$

$$z_{ik} \in \{0,1\} \qquad \forall\, i \in V,\, k \in K \tag{15}$$

(1) represents the objective function value z(S) of a solution s, i.e. the total cost of all tours which has to be minimized. Constraints (2) ensure that every positive supply is assigned to exactly one tour. The following constraints ensure that a supply can only be assigned to a tour which visits the respective location (3) and to which the respective product type has been assigned (4). Constraints (5) guarantee that the depot is included in every tour. Furthermore, constraints (6) ensure that the maximum number of available vehicles is not exceeded. Constraints (7) represent the node degree constraints, guaranteeing that every location that is visited by a vehicle is also left by that vehicle again. Constraints (8) and (9) ensure that the maximum number of compartments per vehicle and the vehicle capacities, respectively, are not exceeded. Constraints (10) are the subtour elimination constraints and constraints (11) – (15) represent the variable domains.

The MCVRP-CFCS extends the CVRP by regarding multiple product types and multiple compartments with flexible sizes. Because the CVRP is already NP-hard, the MCVRP-CFCS is also NP-hard (see e.g. Toth and Vigo 2014).

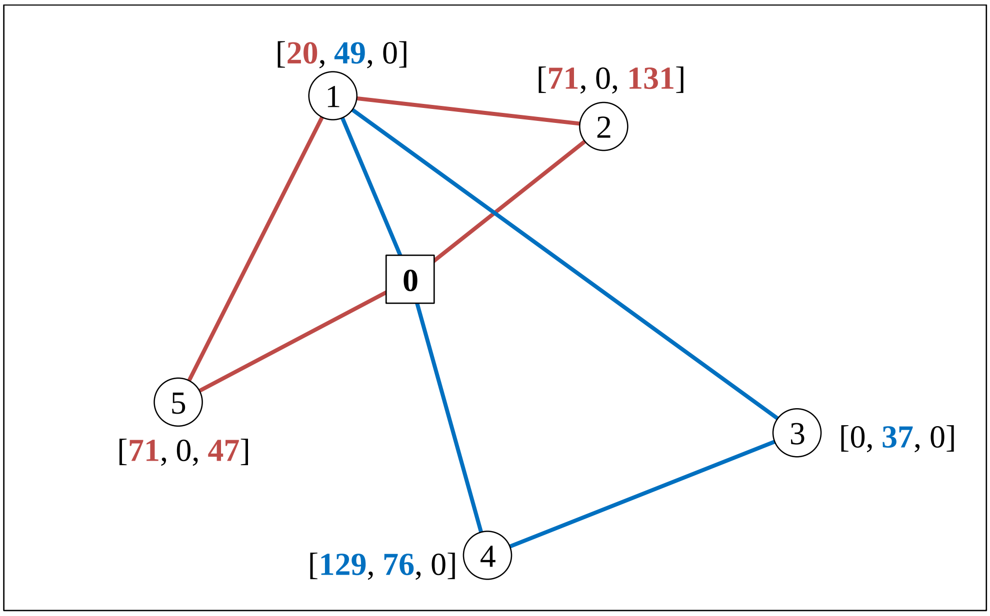


Figure 1: Solution to an example of the MCVRP-CFCS

Figure 1 shows an example for the MCVRP-CFCS with five customer locations, three product types and two available vehicles. Vertex 0 represents the depot; the other vertices represent the customer locations. The supplies for the different product types 1, 2 and 3 are given in square brackets. Each vehicle has a capacity of 350 units and can transport at most two different product types. The figure shows a feasible solution for the problem in which vehicle 1 (indicated by a red line) has compartments for product types 1 and 3 and which collects all supplies (i.e. of both product types) at locations 2 and 5 as well as the supply of product type 1 at location 1. Vehicle 2 (indicated by a blue line) has compartments for product types 1 and 2 and collects all supplies at locations 3 and 4 and the supply of product type 2 at location 1. The example further demonstrates that some locations, e.g. location 2, are only visited once whereas other locations, e.g. location 1, are visited multiple times.

3 Literature Review

One of the first studies about the MCVRP was published by Brown and Graves (1981). They present the case of an American oil company that aims at reducing its transportation costs for the distribution of different fuel products nationwide across the USA. In contrast to the MCVRP-CFCS discussed here, in the problem presented by Brown and Graves the compartment sizes are fixed, the vehicles are not loaded at a central depot but at more than 80 terminals, and (for technical reasons) they cannot deliver the same product type to more than one customer. The problem is formulated as a linear optimization problem and solved as such. However, due to nonsatisfying computing times of the optimal method, a heuristic was developed in order to achieve good solutions within fractions of a second.

Further applications of vehicle routing problems with multiple compartments were presented in the literature, later. Chajakis and Guignard (2003) describe a MCVRP for the transportation of food to convenience stores. The problem is formulated both as a binary and as a mixed-integer linear programming model and solved with a heuristic approach based on Lagrangian relaxation.

Avella et al. (2004) present another MCVRP in the context of petrol replenishment. In this problem, the compartment sizes are fixed and the compartments must be either completely empty or completely filled. The problem is formulated as a set partitioning problem and solved with a heuristic and a branch-and-price-algorithm.

Coelho and Laporte (2015) introduce a classification scheme for MCVRPs in fuel distribution with regard to two aspects. They distinguish between the cases in which the load of one compartment can or cannot be delivered to more than one customer, and the cases in which the demand for a single tank at a customer location can or cannot be delivered by multiple vehicles. They also propose a branch and cut algorithm which solves all four problem variants. According to this classification, the MCVRP-CFCS considered in this paper is a compartment split, tank unsplit problem.

Repoussis et al. (2007) introduce a MCVRP with a heterogeneous fleet and time windows for the depot. The problem is solved with a hybrid metaheuristic consisting of a GRASP algorithm and a variable neighbourhood search. El Fallahi et al. (2008) and Muyldermans and Pang (2010) consider MCVRPs with fixed compartment sizes and where each compartment is assigned a priori to a product type. El Fallahi et al. (2008) solve the problem with a tabu search procedure and a memetic algorithm. Muyldermans and Pang (2010) use a guided local search algorithm. A similar problem is discussed by Mendoza et al. (2010, 2011). However, in contrast to the problem variants studied earlier, demands are considered to be stochastic. The problem is solved by use of construction heuristics and a memetic algorithm. Archetti et al. (2016) compare multi- and single-commodity vehicle routing problems, and provide insights on relationships between the split delivery vehicle routing problem and the MCVRP. Lahyani et al. (2015) present a case study of the collection of different types of olive oil in Tunisia, which represents an extended MCVRP with multiple periods and the possibility of cleaning activities. For this problem, they propose a branch and cut algorithm.

As mentioned above, MCVRPs with flexible compartment sizes are rarely considered yet. One of the few exceptions is the study published by Derigs et al. (2011). They introduce a model which covers both fixed and flexible compartment sizes. Unlike other problems, a customer may further have multiple demands for the same product type. Thus, the same product type can be delivered to a customer by more than one vehicle.

Henke et al. (2015) introduce a MCVRP with discretely flexible compartment sizes. They developed a variable neighbourhood search procedure to solve this problem.

In addition, MCVRPs also occur in a number of maritime transportation problems where liquid or bulk products have to be transported (see e.g. Fagerholt and Christiansen 2000, Jetlund and Karimi 2004, Al-Khayyal and Hwang 2007).

4 Genetic Algorithm

4.1 Overview

In order to solve the MCVRP-CFCS, a genetic algorithm (GA) has been developed. The concept of GAs was first introduced by John H. Holland (1975) and later adapted and applied to various optimization problems. GAs are inspired by natural processes where, over the course of several generations, organisms evolve in order to be better adapted to environmental conditions.

```
procedure GENETIC ALGORITHM (in: instance data, parameters, out: best solution S_best)
  generate (initial) population Pop and determine S_best
  while termination criterion is not met do
     select first parent P_1 ∈ Pop
     apply crossover with probability p_co
     if crossover is applied then
        select second parent P_2 ∈ Pop, P_2 ≠ P_1
        C := CROSSOVER(P_1, P_2)                 // apply crossover and generate child
        apply swap with probability p_swap                         // mutation
        if swap is applied then C := SWAP(C) end if
        apply inversion with probability p_inv
        if inversion is applied then C := INVERSION(C) end if
     else
        C := P_1
        repeat                                                      // mutation
           apply swap with probability p_swap
           if swap is applied then C := SWAP(C) end if
           apply inversion with probability p_inv
           if inversion is applied then C := INVERSION(C) end if
        until at least one mutation is applied
     end if
     C := LOCAL IMPROVEMENT(C)
     determine individual to be removed S              // update the population
     Pop := Pop \{S}, Pop := Pop ∪{C}
     if f(C) < f(S_best) then S_best := C end if
  end while
end procedure
```

Figure 2: Genetic algorithm

In the GA presented here, a population consisting of several individuals (solutions of the underlying optimization problem) is considered. Starting from an initial population, which is generated by specific construction methods, in each generation, i.e. iteration, two individuals of the population are selected as parents by means of a selection rule. With certain probabilities, these parents generate one offspring by application of a crossover operator, a mutation operator or both. In this way, the inheritance of properties of "good" parents should lead to the generation of a "good" offspring. If the offspring fulfils certain criteria with respect to its solution structure and its objective function value, it replaces one individual in the population and the process is repeated by selecting two new parent individuals. This procedure is continued until a pre-defined termination criterion is fulfilled. It is illustrated in Figure 2. In the following subsections, the components of the GA are explained in detail.

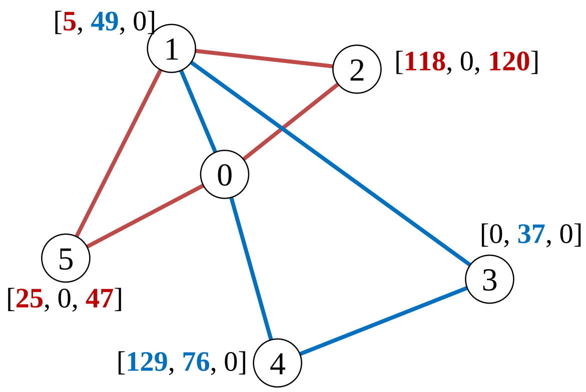
4.2 Representation and Evaluation of a Solution

In a genetic algorithm, a solution is represented as a chromosome consisting of several genes. Within a chromosome, all properties of the solution are encoded. For routing problems, a permutation is often chosen as representation, where each gene represents a customer location and where the order of the genes indicates the sequence in which the locations are visited (see e.g. El Fallahi et al. 2008, Talbi 2009). In the MCVRP-CFCS, however, not only the visiting sequence but also the allocation of supplies to vehicles and the respective compartment sizes for the different product types need to be indentifiable from a chromosome. For this purpose, a representation, which is based on the works of El Fallahi et al. (2008) and Pereira et al. (2002), is used. Each gene represents a positive supply of a specific product type at a specific customer location. The supplies are numbered lexicographically according to the location index first and to the product type index second. Moreover, a chromosome does not consist of a single permutation, but of several strings containing ordered subsets of genes. Each of these subsets represents a tour. Let ρ be the number of positive supplies ($s_{ip} > 0$, $\rho \leq n \cdot |P|$), then a chromosome consists of ρ genes and up to $|K|$ strings. Figure 3 depicts the chromosome of the solution illustrated in Figure 1. It consists of the two subsets of genes that are shaded in grey; the information for customer locations and product types depicted below the actual genes is only given for a better understanding. The sequence in which the supplies are collected and, thus, the sequence in which the customer locations are visited, can be derived from the sequence in which the genes are arranged in the chromosome.

The quality of a solution, i.e. its objective function value, is determined by the total cost of all tours in this solution: The lower the cost, the better is the solution. In a GA, however, an individual is traditionally evaluated by a fitness function where higher values indicate better solutions (see e.g. Talbi 2009). Thus, the objective function value z(S) of a solution S is to be transformed into a fitness value f(S) by application of the following function, in which c_{max} represents the highest cost among all edges:

$$f(S) = n \cdot |K| \cdot c_{max} - z(S)$$

Apart from guaranteeing that a solution with a lower cost is associated with a higher fitness value, the fitness function also needs to ensure that the fitness value is never negative. This is achieved by including the first term $n \cdot |K| \cdot c_{max}$, which is an approximation of the total cost in the worst case.



Tour 1	**8**	**9**	**1**	**3**	**4**
Location	*5*	*5*	*1*	*2*	*2*
Product	*1*	*3*	*1*	*1*	*3*

Tour 2	**2**	**5**	**6**	**7**
Location	*1*	*3*	*4*	*4*
Product	*2*	*2*	*1*	*2*

Compartment sizes:

		Product Type		
		1	**2**	**3**
Tour	**1**	148	0	167
	2	129	162	0

Figure 3: An example of a chromosome

4.3 Initial Population

An important design decision for a GA is the generation of the initial population. On the one hand, the initial population should contain sufficiently diverse solutions in order to avoid premature convergence towards a local optimum. On the other hand, the quality of the solutions in the initial population should be as high as possible in order to quickly generate better solutions. However, this might also lead to a loss of diversity of the population. To incorporate both ideas into the algorithm, one part of the initial population is generated by a completely randomized procedure whereas the other part is generated by a mainly deterministic construction procedure, namely the *savings heuristic* of Clarke and Wright (1964).

In both procedures all (positive) supplies are initially assigned randomly to vehicles. More precisely, in each step a supply that has not been assigned yet is selected at random and assigned to the vehicle with the lowest possible index. It can be assigned to a vehicle if a compartment for the respective product type has already been opened or if it can still be opened, and if the capacity of the vehicle is not exceeded by adding this supply. If a supply cannot be assigned to any of the $|K|$ vehicles, an additional artificial vehicle with capacity Q is considered. In order to account for the resulting infeasibility of the solution, every additional vehicle is penalized by adding a high value to the total cost of the solution resulting in a high reduction of the individual's fitness value.

For the first part of the population, the routes and, consequently, the chromosomes are mainly derived from the sequence in which the supplies were assigned to the tours. That is, if a supply of customer location i is assigned to a tour ahead of a supply of customer location j, then location i is visited before location j and the chromosome is built accordingly. If multiple supplies of the same location are assigned to the same vehicle, then the respective customer location is not visited multiple times. Hence, all genes of the location are positioned consecutively after the first gene corresponding to this location. An example of how a solution is generated randomly is illustrated in Table 1. (The different colors represent the different customer locations.)

Step	Supply				$\hat{k}$	Tour # 1			Tour # 2		
	Gene	**Cust.**	**Prod.**	**Qty.**		**String**	**Comp.**	$\hat{Q}$	**String**	**Comp.**	$\hat{Q}$
0								350			350
1	1	1	1	20	1	1	1	330			350
2	4	2	3	131	1	1 4	1, 3	199			350
3	7	4	2	76	2*	1 4	1, 3	199	7	2	274
4	8	5	1	71	1	1 4 8	1, 3	128	7	2	274
5	2	1	2	49	2*	1 4 8	1, 3	128	7 2	2	225
6	5	3	2	37	2*	1 4 8	1, 3	128	7 2 5	2	188
7	6	4	1	129	2**	1 4 8	1, 3	128	7 6 2 5	1, 2	59
8	3	2	1	71	1	1 4 3 8	1, 3	57	7 6 2 5	1, 2	59
9	9	5	3	47	1	1 4 3 8 9	1, 3	10	7 6 2 5	1, 2	59

Table 1: Example for the generation of a solution for the initial population

(Cust.: customer, Prod.: product type, Qty.: quantity, Comp.: compartment, $\hat{k}$: assigned tour, $\hat{Q}$: remaining capacity;

*Assignment to tour # 1 not possible because the maximum number of compartments is limited, **assignment to tour # 1 not possible because of the capacity constraint)

For the second part of the population, the routes (and, thus, the chromosomes, too) are not constructed according to the chronological assignment of the supplies. Instead, the routes are constructed by application of the well-known savings heuristic of Clarke and Wright (1964). This can easily be done, since the supplies were already assigned to tours randomly in the previous step. The customers assigned to one tour initially form auxiliary single-stop tours. These tours are then merged by using the savings heuristic until a single route is formed. In each iteration of the heuristic, the customer pair (i,j) which corresponds to the highest savings $sav_{ij} = c_{0i} + c_{0j} - c_{ij}$ is identified and the corresponding edge between these customers is added to the route if both customers have at least one connection to the depot.

Finally, a solution is evaluated and included in the initial population if it is not a duplicate of another solution already existing in the initial population. For the identification of duplicates, a simple approach is used here: Two solutions are regarded as duplicates if they have the same fitness value. Although solutions can be wrongly identified as duplicates if they consist of the same routes with a different allocation of compartments or individual supplies, or if they consist of different routes with similar costs, pre-tests have shown that this approach leads to better solutions than an exact search for duplicates. Presumably, this is the case because two solutions with the same fitness value might differ only slightly, e.g. only one supply is assigned differently. If all these very similar solutions were included in the population, there would be a risk of a very low diversity within the population and, thus, of premature convergence.

4.4 Selection Operator

The selection operator determines how parent individuals are chosen for a crossover or a mutation. For the GA described in this paper, this is done by means of the so-called tournament selection (see e.g. Talbi 2009): Two individuals are randomly selected from the population and the one with the higher fitness is chosen as a parent. The second parent is selected in the same way. It has to be ensured, though, that the first parent is not allowed to be selected as the second parent, too. Based on this selection method, the probability for an individual to become a parent depends on its fitness value, i.e. individuals with higher fitness values have a higher probability to be selected as a parent.

4.5 Crossover Operator

With a certain crossover probability, a crossover between the selected individuals is performed. In the literature, crossover operators are frequently used, in which genes are exchanged between the two parents. In contrast, we use an approach in which genetic material is donated from one parent P_1 to the other one P_2, and which is based on the crossover operator proposed by Pereira et al. (2002) for the CVRP. Let K_1 and K_2 be the sets of actually used vehicles in the solutions of P_1 and P_2, i.e. $|K_1| \leq |K|$ and $|K_2| \leq |K|$. Firstly, a tour $t_1 \in K$ from P_1 is chosen randomly. The donated part of P_2 will be inserted (if possible) into this tour. If not all $|K|$ available vehicles are used in the solution represented by P_1 ($|K_1| < |K|$), a tour which is not used yet is selected with a probability of $1 / (2 \cdot |K_1|)$. Hence, the probability of opening a new tour is inversely proportional to the number of already used tours (Pereira et al. 2002). In this way, the option of opening a new tour is assured while it is simultaneously avoided to use an unnecessarily high number of vehicles.

Secondly, a tour $t_2 \in K_2$ from the donating parent P_2 is selected randomly. This tour must be chosen from the set of actually used tours in order to guarantee that it contains genetic material. Then, from the string representing t_2, a substring is selected randomly which is to be donated to P_1. Since the inclusion of all genes of the selected substring into tour t_1 might lead to violations of certain constraints, it needs to be determined whether the whole substring or only some parts of it can be inserted into the genetic material of P_1 by regarding three constraints more closely. (i) The genes of a product type have to be excluded, for which no compartment is available in t_1 and for which no compartment can be opened anymore. (ii) From the first to the last gene of the substring it needs to be checked whether the supplies represented by these genes can be added to tour t_1 without exceeding the respective vehicle's capacity. If adding a certain supply would exceed the capacity, the substring is only selected up to this supply, regardless of subsequent supplies which might have lower supplies. For example, let $\{g_1, \ldots, g_j, \ldots, g_m\}$ be a substring of length m. If the supply represented by gene g_j was the first one exceeding the vehicle's capacity (i.e. $\sum_{i=1}^{j-1} s_{loc(g_i),prod(g_i)} \leq Q$ and $\sum_{i=1}^{j} s_{loc(g_i),prod(g_i)} > Q$), the finally donated substring would be $\{g_1,\ldots,g_{j-1}\}$. (iii) In order to avoid identical genes within a chromosome, all genes included in the substring have to be removed from the genetic material of P_1. Finally, the remaining substring is inserted into t_1 behind the last gene of the location in t_1 that is closest to the first location in the substring.

Figure 4 gives an example illustrating the crossover operator ($n = 5$, $p = 3$, $\widehat{m} = 2$, $\rho = 15$). As before, different customer locations are indicated by different colors. As can be seen, $t_1 = 2$ and $t_2 = 1$ were determined and the substring to be donated is initially {3,4,10,9}. These genes represent supplies of the product types 1 and 3. For both product types there are compartments available in t_1. Hence, only the capacity constraint needs to be checked. Since the gene with value 4 is already contained in t_1, checking the capacity constraint is not necessary at this stage. Furthermore, assuming that the supplies with the gene values 3 and 10 can be added to the tour without exceeding the vehicle capacity, and that the supply represented by gene 9 would exceed the capacity, the final substring that will be donated is {3,4,10}. The respective genes are removed from the present genetic material of P_1 and the substring is inserted behind a gene belonging to the location closest to location 1. In the example, location 1 is already contained in t_1. Thus, the substring can be inserted after the gene of this location. The resulting solution is the current offspring C. As can be seen, the operator allows for adding (by selecting a tour t_1) and removing tours (by removing duplicate genes) from a solution.

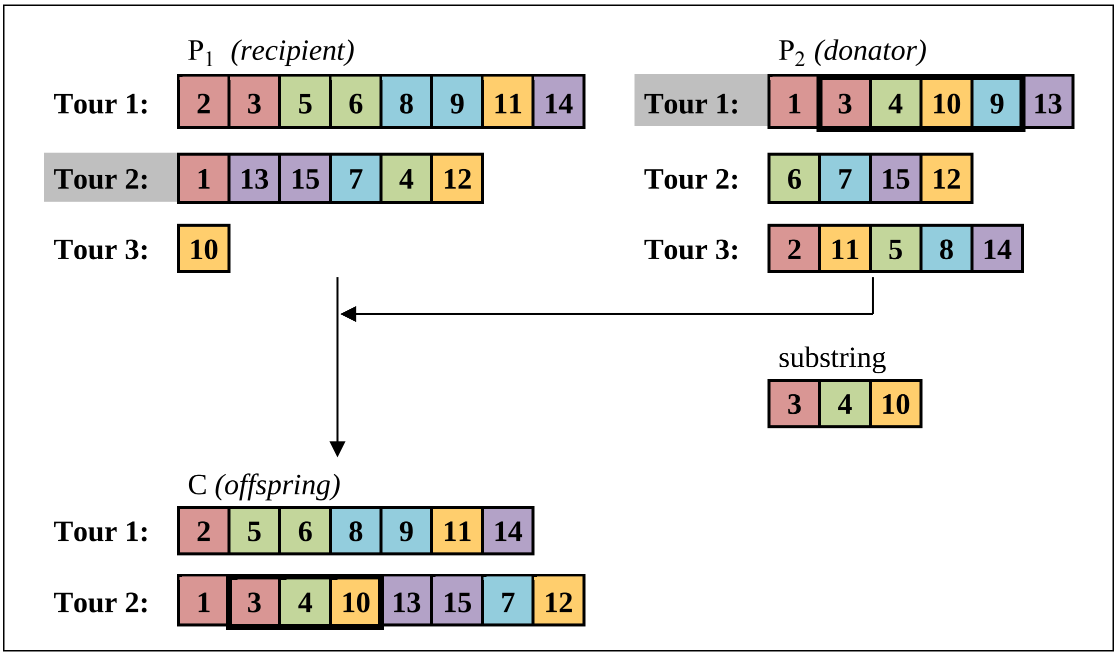


Figure 4: Example of a crossover operation

After applying the crossover operator, the use of an additional repair operator might be necessary. This is the case if an offspring resulted from the crossover in which one or more customer locations are visited more than once within one tour. In the example presented in Figure 4, customer location 4 (genes 10 and 12) is visited twice by vehicle 2. On the one hand, this violates constraint (7). On the other hand, this solution is always dominated by solutions in which location 4 is visited once – either between locations 2 and 5 (genes 4 and 13) or between location 3 (gene 7) and the depot – if the triangle inequality is fulfilled. The repair operator corrects such multiple visits by shifting the respective genes. With respect to the associated costs, it is checked whether it is better to visit customer location 4 between 2 and 5 or between 3 and the depot. Assuming that the first option is chosen, the resulting offspring is shown in Figure 5.

4.6 Mutation Operators

Mutations cause small, random changes to chromosomes in order to help preserving the diversity within a population. In the presented GA, two different mutation operators are considered: a swap operator and an inversion operator.

In the swap operator, two genes are randomly chosen and – if possible – swapped. A swap is possible if capacity and compartment constraints are not violated. It might be necessary to apply the repair operator afterwards.

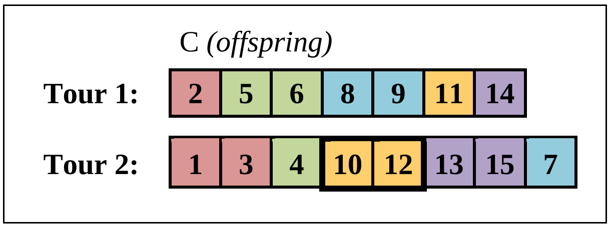


Figure 5: Offspring after the application of the repair operator

In the inversion operator, a tour and a substring within this tour are randomly chosen. Then, the sequence of the genes within this string is reversed. In order to avoid using the repair operator afterwards, the string is expanded if the selected substring would separate two genes representing the same customer location.

If a crossover was performed before, both operators will be applied to the offspring with low probabilities p_{swap} and p_{inv}. Mutation probabilities are typically chosen from the range of 0.001 to 0.05 (Talbi 2009, Srinivas and Patnaik 1994). If the crossover operator was not used, at least one of the mutation operators is used with certainty on solution P_1. In this case, the probability for each operator to be chosen depends on the ratio of p_{swap} and p_{inv}.

4.7 Local Improvement

The local improvement phase is applied after crossover and mutation and consists of two parts. The first part considers a local optimization operator which is applied (only) to the tours affected in the current iteration, i.e. the tour selected as t_1 in the crossover operator and/ or the tour(s) changed by the mutation procedure(s). The operator consists of intra-route shifts, i.e. customers are moved to other positions within the same tour. In each iteration, the shift move with the highest resulting savings in the total cost is performed. For a specific tour, the procedure stops when no further improvement can be achieved.

```
procedure LOCAL IMPROVEMENT (in: instance data, inout: solution S)
  T := set of all tours in S that were affected in the current iteration
  for all tours t ∈ T do
     improve := true             // improve = true, if an improvement was found
     while improve = true do
        c* = cost(t)                 // cost of the best shift
        improve := false
        for i := 1 to n_t do         // for all customers i in tour t
           for p := 1, p ≠ i to n_t do       // for all positions p
              c_shift := cost of tour t if i is shifted to position p
              if c_shift < c* then
                 c* := c_shift, i* := i, p* := p, improve := true
              end if
           end for
        end for
        if improve = true then
           shift customer i* to position p*, cost(t) := c*
        end if
     end while
  end for
  improve := false, count := 1
  repeat
     g := randomly selected gene
     t_g := tour containing g, i := customer corresponding to g
     generate a substring sstr with all genes of i in t_g
     repeat
        t := randomly selected tour, t ≠ t_g
        if t contains i and all product types in sstr and sstr can be added feasibly to t then
           shift sstr to t, improve := true
        end if
     until all tours except t_g have been tested or improve = true
     count := count + 1
  until improv = true or count = shift_max
end procedure
```

Figure 6: Local improvement

In the second phase, in order to improve the solution further, inter-route shifts of all supplies of a customer are applied (if possible) from one tour to another tour in which the respective customer is also visited. That way, the visit of a location could be completely removed from one tour and, thus, the solution could be improved. More precisely, a gene is randomly selected from all tours and a substring is formed containing the selected gene and all genes corresponding to the same customer in the same tour. The procedure then examines a shift of this substring to another randomly selected tour in which the selected customer is also visited, the same product types are transported and the capacity constraint would not be violated. The improvement procedure is repeated until one change in the solution was made or until a given number of shifts has been examined. Both procedures are depicted in Figure 6.

4.8 Population Reinitialization

If no improvement of the best solution was achieved after a certain number of iterations, the population is (partly) reinitialized in order to lead the search towards other parts of the solution space. This is done by replacing the worst solutions in the current population by new, randomly generated solutions. The new solutions are obtained in the same manner as the solutions in the initial population (see section 0). The percentage of solutions to be replaced in the current population is determined by a predefined parameter. The reinitialization is triggered after *iter_reinit* iterations without improvement or after *iter_reinit* iterations after the last reinitialization.

4.9 Replacement Strategy and Termination Criteria

As mentioned in Section 4.1, the newly determined offspring should replace an individual from the current population, i.e. the population is changing continuously throughout the procedure. This procedure corresponds to the so-called *steady state model* (see e.g. Talbi 2009). An offspring is only included in the population if it is not a duplicate of an already existing solution and if its fitness value is higher than the currently lowest fitness value in the population. It replaces a randomly chosen individual with a fitness value below the average fitness value in the population.

The algorithm stops if a certain number of iterations were performed or if no improvement of the currently best solution was achieved for a certain number of iterations.

5 Numerical Experiments

5.1 Overview

The presented GA was implemented in Microsoft Visual Studio C++ 2015 and the experiments were performed on a 3.07 GHz and 4 GB RAM computer.

Extensive numerical experiments were conducted which can be divided into three parts. In the first part, the economic advantage of using continuously flexible compartments is investigated, whereas in the remaining two parts the performance of the GA with respect to solution quality and computing time is analysed by solving small instances with 10 locations on the one hand and larger instances with up to 50 locations on the other hand. The size of the small instances allowed for computing the optimal solutions for a large number of instances within acceptable computing times by using the branch-and-cut approach developed by Henke et al. (2017). In this way, it was possible to evaluate the solution quality of the GA for varying problem specifications in-depth by comparing the heuristic solutions to the optimal solutions. It was not possible to compare the GA with existing algorithms because problem variants solved by other algorithms are different from the MCVRP-CFCS. Therefore, further experiments on a small number of larger instances have been conducted in order to analyse the performance of the GA for practically relevant problem sizes. Instances introduced by Henke et al. (2015, 2017) were used for the experiments.

5.2 Problem Instances

In the first instance set, the so-called small instances (Henke et al. 2015), ten customer locations are considered in each instance. They differ with respect to the number of product types ($|P| \in \{3,6,9\}$), the maximum number of compartments ($\hat{m} \in \{2,3\}$ for $|P| = 3$, $\hat{m} \in \{2,4,6\}$ for $|P| = 6$, $\hat{m} \in \{2,4,7,9\}$ for $|P| = 9$) and a parameter indicating the number of supplies $\bar{s}$ at the customer locations. For this parameter, three options are considered: small ($\bar{s} = 1$), medium ($\bar{s} = 2$), and large ($\bar{s} = 3$). For a small number of supplies, a supply for at least one but at most one third of the product types occurs at each customer location. For a medium level, each customer location has a supply for more than one third and at most two thirds of the product types. Finally, for a large number of supplies, each customer location provides more than two thirds of the product types. Thus, the instances can be divided into 27 problem classes with each class containing 50 different instances.

The second set of instances, the so-called large instances (Henke et al. 2017), consists of 50 instances with 10 to 50 customers, three or four product types, a varying maximal number of compartments ($\hat{m} \in \{2,3\}$ for $|P| = 3$, $\hat{m} \in \{2,3,4\}$ for $|P| = 4$) and a varying number of supplies which is defined similarly to the small instances. The considered instances are a subset of some of the instances introduced by Henke at al. (2017) for which their branch-and-cut approach was able to identify optimal solutions within less then 10 minutes of computing time.

5.3 Comparison between Discretely and Continuously Flexible Compartments

While in the MCVRP-CFCS the compartment sizes can be varied continuously, Henke et al. (2015, 2017) considered a problem variant with discretely flexible compartment sizes. That is, the compartment sizes can be varied only in equal steps based on a basic compartment unit size q^{unit}. This problem variant is motivated by a real-world application in the context of waste glass collection. In this problem, the vehicle capacity is divided by separation walls which can only be inserted at specific positions. Hence, the solution space is further restricted compared to the problem with continuously flexible compartment sizes. Consequently, objective function values of the MCVRP-CFCS cannot be higher than those obtained in the MCVRP with discretely flexible compartment sizes. In order to investigate how large the difference in the objective function values can be and under which problem characteristics continuously flexible compartments are especially beneficial, we compare the optimal objective function values for the problem variant with continuously flexible compartment sizes and two cases of the problem variant with discretely flexible compartments sizes. In the first discrete case, a q^{unit} was set to 0.05Q, i.e. the size of a compartment must be an integer multiple of 5% of the vehicle capacity. Analogously, for the second discrete case a q^{unit} of 0.1Q was selected. In order to obtain the optimal values for all three cases, the 1,350 small instances have been solved to optimality by applying the branch-and-cut algorithm of Henke et al. (2017), respectively.

parameters			optimal objective function values			deviation from z.cont	
|*P*|	$\hat{m}$	$\bar{s}$	**z.cont**	**z.discr-0.05**	**z.discr-0.1**	**dev.discr-0.05**	**dev.discr-0.1**
3	2	1 (small)	363.80	365.18	366.90	0.38%	0.86%
		2 (medium)	464.68	464.89	465.35	0.04%	0.13%
		3 (large)	596.16	596.16	596.16	0.00%	0.00%
	3	1 (small)	343.32	344.99	349.08	0.48%	1.76%
		2 (medium)	332.33	333.88	337.15	0.51%	1.65%
		3 (large)	332.94	334.43	336.64	0.45%	1.11%
6	2	1 (small)	549.98	549.98	549.98	0.00%	0.00%
		2 (medium)	767.48	767.49	767.48	0.00%	0.00%
		3 (large)	919.78	919.78	919.78	0.00%	0.00%
	4	1 (small)	361.67	364.22	366.98	0.69%	1.49%
		2 (medium)	491.39	491.49	492.11	0.02%	0.15%
		3 (large)	583.18	583.18	584.46	0.00%	0.18%
	6	1 (small)	334.50	338.18	351.98	1.12%	5.31%
		2 (medium)	334.96	342.12	357.98	2.17%	6.92%
		3 (large)	335.21	342.87	357.42	2.29%	6.67%
9	2	1 (small)	724.17	724.17	724.17	0.00%	0.00%
		2 (medium)	1,114.25	1,114.25	1,114.25	0.00%	0.00%
		3 (large)	1,419.09	1,419.09	1,419.09	0.00%	0.00%
	4	1 (small)	461.78	461.78	463.05	0.00%	0.29%
		2 (medium)	646.41	646.50	647.57	0.01%	0.18%
		3 (large)	815.27	815.27	815.27	0.00%	0.00%
	7	1 (small)	341.17	347.09	360.16	1.73%	5.58%
		2 (medium)	449.27	450.21	456.97	0.22%	1.71%
		3 (large)	556.35	556.35	564.50	0.00%	1.52%
	9	1 (small)	321.28	333.56	354.98	3.78%	10.39%
		2 (medium)	345.78	359.36	392.48	3.94%	13.63%
		3 (large)	328.28	342.10	383.36	4.27%	16.95%
minimum			321.28	333.56	336.64	0.00%	0.00%
average			550.48	552.79	558.63	0.69%	2.39%
maximum			1,419.09	1,419.09	1,419.09	4.27%	16.95%

Table 2: Average objective function values per problem class for the MCVRP with continuously and discretely flexible compartment sizes and average deviations from the MCVRP-CFCS

Table 2 contains the optimal objective function values of the 1,350 problem instances (aggregated into problem classes) for the three variants of the MCVRP with continuously flexible compartment sizes (z.cont), with discretely flexible compartment sizes and q^{unit}=0.05Q (z.discr-0.05), and q^{unit}=0.1Q (z.discr-0.1). In addition, the average deviations of the objective function values with q^{unit}=0.05Q (dev.discr-0.05) and q^{unit}=0.1Q (dev.discr-0.1) from the objective function values in the continuous case are given. It can be observed that these values differ by 0.69% between the continuous case and the discrete case with q^{unit}=0.05Q, and even by 2.39% between the continuous case and the discrete case with q^{unit}=0.1Q. The results demonstrate that discreteness of compartment sizes has an impact on the total cost. This impact becomes larger with an increasing basic compartment unit size. An explanation for this finding can be given as follows: Due to the discreteness of the compartments, a part of a compartment's capacity might remain unused because the sum of the volumes of the assigned supplies does not equal a feasible compartment size. An increased unit capacity q^{unit} may result in a larger amount of unused capacity.

The results further show that the deviation of the objective function values between the three problem cases seems to be dependent on the problem parameters. The average deviations increase significantly with an increasing number of product types and an increasing number of compartments for both discrete cases. However, with respect to the number of supplies the results are ambiguous.

Based on these results, it can be concluded that continuously flexible compartments are especially beneficial where problems with large numbers of product types and compartments have to be considered. The results further show that, on average, a larger discretization leads to higher total costs.

5.4 Experiments with Small, Randomly-generated Instances

In pre-tests, 50 out of the 1,350 small instances were selected randomly in order to determine the best parameters for the GA. For this purpose, the algorithm was applied 1,000 times to each of the 50 instances and for different parameter settings. Consequently, the following parameters for the GA were determined:

- population size σ = 100;
- share of solutions in the initial population for which the savings heuristic is applied π_{sav} = 0.6;
- share of solutions in the population to be replaced in the case of reinitialization π_{reinit} = 0.4;
- crossover probability p_{co}= 0.9;
- mutation probabilities p_{swap}= 0.05 and p_{inv}= 0.05;
- number of iterations after the last improvement or the reinitialization that triggers reinitialization *iter_reinit* = 500;
- maximum number of attempts for the inter-route shift operator of the local improvement phase *shift_max* = 50

In the experiments, the GA was applied to all 1,350 instances. Each instance was solved five times with the GA to obtain reliable results. The experiments were conducted with computing time limits of one (GA-1) and five seconds (GA-5), respectively. The results of the experiments are given in Table 3 for both computing time limits. For each problem class and computing time limit, it is shown how many instances could be solved to optimality (*#opt*) by the GA. An instance is considered as solved to optimality by the GA if the optimal solution was found in at least one of the five runs. In addition, the columns *avg.opt* indicate in how many of the five runs the optimal solutions were found on average.

The columns *avg.err* and *max.err* show the average and the maximum deviations, respectively, of the average objective function values obtained by the GA from the optimal objective function values.

parameters			**GA-1**				**GA-5**			
|*P*|	**$\hat{m}$**	**$\bar{s}$**	**#opt**	**avg.opt**	**avg.err**	**max.err**	**#opt**	**avg.opt**	**avg.err**	**max.err**
3	2	1	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
		2	49	4.6	0.04%	2.58%	50	4.7	0.01%	2.58%
		3	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
	3	1	50	4.9	0.00%	0.00%	50	4.9	0.00%	0.00%
		2	50	5.0	0.00%	0.01%	50	5.0	0.00%	0.01%
		3	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
6	2	1	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
		2	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
		3	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
	4	1	50	4.9	0.00%	0.00%	50	4.9	0.00%	0.00%
		2	48	4.3	0.11%	5.25%	50	4.4	0.07%	2.18%
		3	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
	6	1	50	4.9	0.00%	0.00%	50	4.9	0.00%	0.00%
		2	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
		3	50	4.9	0.00%	0.00%	50	4.9	0.00%	0.00%
9	2	1	50	4.7	0.05%	2.28%	50	4.9	0.00%	0.01%
		2	47	3.1	0.38%	5.46%	49	4.0	0.19%	4.35%
		3	49	4.7	0.01%	0.46%	49	4.7	0.00%	0.23%
	4	1	49	4.5	0.16%	3.85%	50	4.6	0.13%	3.85%
		2	45	3.6	0.32%	5.22%	46	3.8	0.25%	5.22%
		3	50	4.7	0.05%	2.34%	50	4.9	0.02%	2.30%
	7	1	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
		2	47	3.7	0.48%	8.00%	49	4.3	0.14%	7.16%
		3	50	4.0	0.21%	4.06%	50	4.6	0.07%	3.66%
	9	1	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
		2	50	5.0	0.00%	0.00%	50	5.0	0.00%	0.00%
		3	50	5.0	0.00%	0.01%	50	5.0	0.00%	0.01%
minimum			45	3.1	0.00%	0.00%	46	3.8	0.00%	0.00%
average			49.4	4.7	0.07%	1.46%	49.7	4.8	0.03%	1.17%
maximum			50	5.0	0.48%	8.00%	50	5.0	0.25%	7.16%

Table 3: Numbers of optimal solutions provided by the GA, and average and maximum deviations from the corresponding optimal objective function values for the problem instances of Henke et al. (2015)

With an average deviation from the optimal objective function values of 0.07 % over all instances, the GA obtained very good results already after a computing time of one second. Within this computing time, the algorithm was able to solve all 50 instances of 20 out of the 27 problem classes at least once to optimality, and in total 1,334 out of the 1,350 instances (98.8 %) were solved optimally in at least one of the five runs. Furthermore, 1,175 instances (87.0 %) of all instances were solved to optimality in all five runs.

With a time limit of five seconds, the average deviation from the optimal objective function values across all instances could be reduced to 0.03 %, optimal solutions were found for 1,343 instances (99.5 %) in at least one run, and for 1,232 instances (91.3 %) in all five runs.

The solution quality depends on different parameters. Naturally, the underlying VRP becomes more difficult to be solved with an increasing number of customers. Since this parameter is kept constant in this set of experiments, the influence of the remaining parameters (number of product types, compartments and supplies) on the performance of the genetic algorithm can be analyzed. For the algorithm, the problem seems to become more difficult to be solved with an increasing number of product types as the average deviations from the optimal objective function values amount to 0.007 %, 0.013 % and 0.139 % for 3, 6 and 9 product types, respectively. Moreover, the solution quality decreases with an increasing maximal number of compartments. However, the case $|P| = \widehat{\mathrm{m}}$ seems to be an exception from this finding as almost all of these instances could be solved optimally. The number of supplies does not seem to have a clear effect on the solution quality. Whereas the results for a small and a large number of supplies are quite similar, the instances with a medium number of supplies show higher deviations from the optimal solutions. These results are similar to the findings of Henke et al. (2015, 2017) with respect to the number of product types, but they are different with respect to the maximal number of compartments and the number of supplies. For a simple exact approach, Henke et al. (2015, 2017) found that a decreasing maximal number of compartments and an increasing number of supplies lead to a significant increase in computing time. For a branch-and-cut approach, Henke et al. (2017) confirm that a decreasing maximal number of compartments leads to a significant increase in computing time whereas their findings with respect to the number of supplies are similar to the impact on the GA. In comparison, the results suggest that the performance of the GA is not as much influenced by these two problem parameters as exact solution approaches are.

5.5 Experiments with Large, Randomly-generated Instances

The results of the experiments with the second set of instances with 10 to 50 customers are given in Table 4. Although a few outliers occurred, the GA obtained very good results for the majority of the instances. With a computing time limit of one second (five seconds), the average deviation from the optimal objective function values amounts to 0.89 % (0.58 %), and for 31 (35) instances the average deviations from the optimal solutions amounted to less than 0.5 %. Optimal solutions were found at least once for almost all instances. Only four instances were not solved optimally within one second. However, by increasing the computing time limit to five seconds, the optimal solutions for three of these instances could be obtained in three or four of the five runs.

It could be observed that the number of customers does not have a clear impact on the performance of the GA with respect to the average deviation from the optimal solutions. As conclusion, it can be observed that even for larger instances the performance of the GA is very good within very small computing times.

no. of customers	no. of instances	GA-1			GA-5		
		#opt	**avg.opt**	**avg.err**	**#opt**	**avg.opt**	**avg.err**
10	2	2	4.0	0.72%	2	5.0	0.00%
15	5	5	4.2	1.48%	5	4.4	1.05%
20	7	7	4.4	0.14%	7	4.9	0.03%
25	5	5	5.0	0.00%	5	5.0	0.00%
30	5	5	5.0	0.00%	5	5.0	0.00%
35	7	7	2.9	1.33%	7	3.4	1.10%
40	8	7	2.9	1.18%	8	3.8	0.45%
45	6	6	3.0	1.02%	6	4.0	0.88%
50	5	2	0.8	1.96%	4	2.8	1.45%
total	50	46	3.5	0.89%	49	4.2	0.58%

Table 4: Numbers of optimal solutions provided by the GA and average deviations from the corresponding optimal objective function values for the problem instances of Henke et al. (2017)

6 Conclusions and Outlook on Future Research

In this paper, a variant of the multi-compartment vehicle routing problem has been presented. Unlike in most of the previously studied problem variants, the compartment sizes can be varied completely freely. Moreover, the number of compartments that can be opened in a vehicle might be limited, i.e. it is possible that a vehicle cannot transport all available product types at the same time. Hence, additional decisions – compared to the classical CVRP – have to be made, namely which compartments are used on the individual tours and which sizes are chosen for these compartments. Due to the complexity of the problem, it can be solved to optimality only for smaller instances within reasonable computing times. Therefore, a genetic algorithm was developed which is based on different genetic algorithms for the CVRP from the literature.

The algorithm was tested on 1,400 instances and with two alternative computing time limits of one and five seconds. As it was to be expected, the solution quality is improving with increasing computing time limit; however, also within only one second, the majority of the instances could be solved to optimality. Moreover, it was shown that the possibility to vary the compartment sizes continuously can lead to significant cost savings compared to discretely flexible compartment sizes, especially in planning situations with a large number of product types and compartments. Although there is not much research dealing with the MCVRP-CFCS yet, these results might serve as an incentive to intensify future research about it. Amongst others, the analysis of problem modifications, e.g. stochastic supplies, and problem extensions, e.g. a multi-periodic context, could provide interesting research opportunities.

References

Al-Khayyal, F.; Hwang, S.-J. (2007): Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, part i: applications and model. In: European Journal of Operational Research 176, 106-130.

Archetti, C.; Campbell, A.M.; Speranza, M.G. (2016): Multicommodity vs. single-commodity routing. In: Transportation Science 50, 461-472.

Avella, P.; Boccia, M.; Sforza, A. (2004): Solving a fuel delivery problem by heuristic and exact approaches. In: European Journal of Operational Research 152, 170-179.

Brown, G.G.; Graves, G.W. (1981): Real-time dispatch of petroleum tank trucks. In: Management Science 27, 19-32.

Chajakis, E.D.; Guignard, M. (2003): Scheduling deliveries in vehicles with multiple compartments. In: Journal of Global Optimization 26, 43-78.

Clarke. G.; Wright, J.W. (1964): Scheduling of vehicles from a central depot to a number of delivery points. In: Operations Research 12, 568-581.

Coelho, L.C.; Laporte, G. (2015): Classification, models and exact algorithms for multi-compartment delivery problems. In: European Journal of Operational Research 242, 854-864.

Dantzig, G.B.; Ramser, J.H. (1959): The truck dispatching problem. In: Management Science 6, 80-91.

Derigs, U.; Gottlieb, J.; Kalkoff, J.; Piesche, M.; Rothlauf, F.; Vogel, U. (2011): Vehicle routing with compartments: applications, modelling and heuristics. In: OR Spectrum 33, 885-914.

El Fallahi, A.; Prins, C.; Wolfler Calvo, R. (2008): A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. In: Computers & Operations Research 35, 1725-1741.

Fagerholt, K., Christiansen, M. (2000): A combined ship scheduling and allocation problem. Journal of the Operational Research Society 51, 834-842.

Henke, T.; Speranza, M.G.; Wäscher, G. (2015): The multi-compartment vehicle routing problem with flexible compartment sizes. In: European Journal of Operational Research 246, 730-743.

Henke, T.; Speranza, M.G.; Wäscher, G. (2017) A branch-and-cut algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. Working Paper No. 04/2017, Faculty of Economics and Management, Otto-von-Guericke University Magdeburg.

Holland, J.H. (1975): Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence. Ann Arbor: University of Michigan Press.

Jetlund, A.S.; Karimi, I.A. (2004): Improving the Logistics of Multi-Compartment Chemical Tankers. In: Computers & Chemical Engineering 28, 1267-1283.

Lahyani, R.; Coelho, L.C.; Khemakhem, M.; Laporte, G.; Semet, F. (2015a): A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. In: Omega 51, 1-10.

Mendoza, J.E.; Castanier, B.; Guéret, C.; Medaglia, A.L.; Velasco, N. (2010): A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. In: Computers & Operations Research 37, 1886-1898.

Mendoza, J.E.; Castanier, B.; Guéret, C.; Medaglia, A.L.; Velasco, N. (2011): Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. In: Transportation Science 45, 346-363.

Muyldermans, L.; Pang, G. (2010): On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. In: European Journal of Operational Research 206, 93-103.

Pereira, F.B.; Tavares, J.; Machado, P.; Costa, E. (2002): GVR: a new genetic representation for the vehicle routing problem. In: O'Neill, M. et al. (eds.): Artificial Intelligence and Cognitive Science. Berlin, Heidelberg: Springer, 95-102.

Repoussis, P.P.; Tarantilis, C.D.; Ioannou, G. (2007): A hybrid metaheuristic for a real life vehicle routing problem. In: Boyanov, T. et al. (eds.): Numerical Methods and Applications, Lecture Notes in Computer Science 4310. Berlin, Heidelberg: Springer, 247-254.

Srinivas, M.; Patnaik, L.M. (1994): Adaptive probabilities of crossover and mutation in genetic algorithms. In: IEEE Transactions on Systems, Man, and Cybernetics 24, 656-667.

Talbi, E. (2009): Metaheuristics: from design to implementation. Hoboken: Wiley.

Toth, P.; Vigo, D. (2014): Vehicle routing: problems, methods, and applications (2nd ed.). Philadelphia: Society for Industrial and Applied Mathematics.

VI

A Matheuristic for the Multi-Compartment Vehicle Routing Problem with Multiple Periods

A Matheuristic for the Multi-Compartment Vehicle Routing Problem with Multiple Periods

Tino Henke

Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany

tino.henke@ovgu.de

M. Grazia Speranza

Department of Quantitative Methods, University of Brescia, 25122 Brescia, Italy

speranza@eco.unibs.it

Gerhard Wäscher

Department of Management Science, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany

School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, 100044 Beijing, China

gerhard.waescher@ovgu.de

Abstract

This paper deals with a multi-compartment vehicle routing problem with multiple periods in which the sizes and the number compartments are flexible for each vehicle. Such a problem occurs, for example, in the context of glass waste collection in which different types of glass waste need to be collected from containers while ensuring that the different types do not get mixed during transportation. For this problem, a mathematical model is introduced, which can be implemented for the determination of feasible or optimal solutions for small problem sizes. In order to solve larger problem instances, a matheuristic has been developed and extensively tested by means of numerical experiments. The results demonstrate that a good solution quality can be obtained within acceptable computing times. Further experiments have been conducted in order to investigate the benefits of taking a multi-period planning horizon into account. The corresponding results indicate that substantial savings can be obtained in comparison to a single-period approach to the problem.

Keywords: vehicle routing, multiple compartments, multiple periods, matheuristic

1 Introduction

Vehicle routing problems were first mentioned by Dantzig and Ramser (1959) and have since been extensively studied (Laporte, 2009, Toth and Vigo, 2014). Especially during the last decade, problems which represent complex real-world applications have been investigated. One such class of problems is known as the multi-compartment vehicle routing problem (MCVRP). In general, respective problems occur in settings in which (1) multiple product types have to be delivered (or to be collected) without being mixed and (2) vehicles with multiple loading areas, i.e. compartments, are available. Among others, such conditions occur in the transportation of liquid, bulk or semi-bulk products, e.g. petroleum products (e.g. Avella et al., 2004), agricultural products (e.g. Lahyani et al., 2015), or waste (e.g. Elbek and Wøhlk, 2016), as well as in the transportation of product types with different transportation requirements, e.g. dry, cooled, and frozen products in the delivery to convenience stores (e.g. Hübner and Ostermeier, 2017).

The application considered in this paper concerns the collection of glass waste from containers. In this context, different types of glass waste, e.g. colorless, green, brown glass, have to be collected by vehicles while ensuring that they do not get mixed during transportation. Until now, this problem has only been studied for the single-period case (Henke et al., 2015). However, in a real-world setting the consideration of multiple periods might allow for a more efficient routing of vehicles and, thus, less operational costs as it might not be necessary to collect glass waste from each container in each period. Deciding about when to empty a container and how to determine routes for the collection vehicles is interdepend to each other. One aim of this paper is to investigate the impact on operational costs of considering multiple planning periods simultaneously. Furthermore, a mathematical model for the problem is presented which allows for solving very small instances to optimality. In order to be able to solve larger problem instances as well, a heuristic solution approach, namely a matheuristic, has been developed. Based on extensive numerical experiments, the model and the matheuristic have been tested and the results will be presented.

The main contributions of this paper are (1) the introduction, formulation and analysis of a multi-period MCVRP, (2) the development of a matheuristic which solves this complex problem with an average deviation from the best known solutions of 1.07%, and (3) the investigation of the benefits of multi-period planning in comparison to single-period planning, which demonstrates that average savings of 4.6% are generated if multiple periods are considered simultaneously.

The remainder of this paper is structured in the following manner: In the next section, the considered multi-period planning problem is described in detail and a mixed integer model for this problem is introduced. In Section 3, a short review on the literature on MCVRPs in general and on MCVRPs with multiple periods in particular is given. The developed matheuristic is explained in detail in Section 4, and the extensive numerical experiments and their corresponding results are presented in Section 5. The paper finishes with a summary of the main conclusions.

2 Problem Description and Formulation

In the following, the MCVRP with multiple periods (MCVRP-MP) will be described. For illustration, the example of glass waste collection will be used. However, the problem description can easily be generalized or adapted to other applications.

In the MCVRP-MP a set of locations *V*, *V* = {0, 1, …, n}, which consists of a depot {0} and some container locations {1, …, n}, and a set of product types *P*, e.g. types of glass waste, are regarded over a set of consecutive planning periods *T*, e.g. weeks. A specific container will be denoted by the index tuple (i, p) indicating a container at location i which contains glass waste of type p. At the beginning of the planning horizon, each container (i, p) has an initial fill level of glass waste l_{ip}^{init}, and over the course of each period t the fill level at the beginning of t l_{ipt} is increased by a constant and deterministic amount f_{ip} of glass waste. This amount f_{ip} will be referred to as the fill rate. It needs to be ensured that a container is fully emptied before its capacity q_{ip} is exceeded. For emptying the containers, a sufficiently large set of homogeneous vehicles *K*, each with total capacity Q, is available at a depot {0}. The capacity of a vehicle can be separated into a limited number of compartments, each of which provides space for the transportation of exactly one type of glass waste. Individually for each period and each vehicle, the separation of the capacity into compartments is completely flexible as long as the total capacity of the vehicle is respected and the number of compartments does not exceed a given maximum number of compartments $\widehat{m}$. Each container must always be emptied completely by a single vehicle. Furthermore, the travelling cost c_{ij} between each pair of locations (i, j) is known and symmetric.

In order to solve the problem, the following decisions have to be made:

- Which container is to be emptied in which period(s), i.e. which collection schedule is to be applied?
- For each period, which container is to be emptied by which vehicle, i.e. which container is to be emptied on which tour?
- For each period and each vehicle, how is the total capacity of this vehicle to be separated into compartments and which types of glass waste are to be assigned to these compartments?
- For each period and each vehicle, in which sequence are the assigned container locations to be visited by the vehicle?

The aim of the problem is to identify a collection schedule and vehicle routes for all periods in *T* which minimize the total cost, i.e. the sum of all costs which occur on the determined routes over the complete planning horizon.

In order to formulate a model for the MCVRP-MP, the sequence of the collection activities and the fill level changes is further specified as follows: The fill level of a container at the beginning of a period is equal to its fill level at the end of the previous period. Subsequently, only at the beginning of a period this fill level may or may not be collected as a supply of glass waste. During a period, another fill rate is then added to the container. Hence, the fill level of a container at the end of a period is computed as its fill level at the beginning of the period minus the collected supply in this period plus the fill rate.

In the model, the following variables are introduced:

l_{ipt} :	fill level of container (i, p) at the end of period t,	$\forall\ i\in V\setminus\{0\}, p\in P, t\in T;$
$u^B_{ipkt} = \begin{cases} 1, \text{ if supply (i, p) is collected by vehicle k in period t,} \\ 0, \text{ otherwise,} \end{cases}$		$\forall\ i\in V\setminus\{0\}, p\in P, k\in K, t\in T;$
u^R_{ipkt} :	quantity collected from container (i, p) by vehicle k in period t,	$\forall\ i\in V\setminus\{0\}, p\in P, k\in K, t\in T;$
$y_{pkt} = \begin{cases} 1, \text{ if any compartment for product type p is used in vehicle k in period t,} \\ 0, \text{ otherwise,} \end{cases}$		$\forall\ p\in P, k\in K, t\in T;$
$x_{ijkt} = \begin{cases} 2, \text{ if i = 0 and edge (i, j) is used twice by vehicle k in period t,} \\ 1, \text{ if edge (i, j) is used once by vehicle k in period t,} \\ 0, \text{ otherwise,} \end{cases}$		$\forall\ i,j\in V, i<j, k\in K, t\in T;$
$z_{ikt} = \begin{cases} 1, \text{ if location i is visited by vehicle k in period t,} \\ 0, \text{ otherwise,} \end{cases}$		$\forall\ i\in V, k\in K, t\in T;$
x_0 :	total cost of a solution.	

The model can then be formulated as follows:

$$\min \sum_{(i,j)\in A} \sum_{k\in K} \sum_{t\in T} c_{ij} \cdot x_{ijkt} ; \qquad (1)$$

$$l_{ipt} = l_{ip,t-1} + f_{ip} - \sum_{k\in K} u^R_{ipkt} , \qquad \forall\ i\in V\setminus\{0\}, p\in P, t\in T\setminus\{1\}; \qquad (2)$$

$$\sum_{k\in K} u^R_{ipkt} \le l_{ip,t-1} , \qquad \forall\ i\in V\setminus\{0\}, p\in P, t\in T\setminus\{1\}; \qquad (3)$$

$$l_{ip1} = l^{init}_{ip} + f_{ip} - \sum_{k\in K} u^R_{ipkt} , \qquad \forall\ i\in V\setminus\{0\}, p\in P; \qquad (4)$$

$$\sum_{k\in K} u^R_{ipk1} \le l^{init}_{ip} , \qquad \forall\ i\in V\setminus\{0\}, p\in P; \qquad (5)$$

$$u^R_{ipkt} \le q_{ip} \cdot u^B_{ipkt} , \qquad \forall\ i\in V\setminus\{0\}, p\in P, k\in K, t\in T; \qquad (6)$$

$$l_{ipt} \le f_{ip} + (q_{ip} - f_{ip}) \cdot (1 - \sum_{k\in K} u^B_{ipkt}) , \qquad \forall\ i\in V\setminus\{0\}, p\in P, t\in T; \qquad (7)$$

$$\sum_{i\in V\setminus\{0\}} \sum_{p\in P} u^R_{ipkt} \le Q , \qquad \forall\ k\in K, t\in T; \qquad (8)$$

$$u^B_{ipkt} \le y_{pkt} , \qquad \forall\ i\in V\setminus\{0\}, p\in P, k\in K, t\in T; \qquad (9)$$

$$\sum_{p\in P} y_{pkt} \le \hat{m} , \qquad \forall\ k\in K, t\in T; \qquad (10)$$

$$u^B_{ipkt} \le z_{ikt} , \qquad \forall\ i\in V\setminus\{0\}, p\in P, k\in K, t\in T; \qquad (11)$$

$$z_{ikt} \le z_{0kt} , \qquad \forall\ i\in V\setminus\{0\}, k\in K, t\in T; \qquad (12)$$

$$\sum_{\substack{j\in V:\\ j>0}} \sum_{k\in K} x_{0jkt} \le 2\cdot|K| , \qquad \forall\ t\in T; \qquad (13)$$

$$\sum_{\substack{j\in V:\\ i<j}} x_{ijkt} + \sum_{\substack{j\in V:\\ j<i}} x_{jikt} = 2 \cdot z_{ikt} , \qquad \forall\ i\in V, k\in K, t\in T; \qquad (14)$$

$$\sum_{i\in S} \sum_{\substack{j\in S:\\ i<j}} x_{ijkt} \leq |S| - 1 , \qquad \forall\ k\in K, t\in T, S\subseteq V\backslash\{0\}, |S|>2; \qquad (15)$$

$$l_{ipt} \geq 0 , \qquad \forall\ i\in V\backslash\{0\}, p\in P, t\in T; \qquad (16)$$

$$u^{R}_{ipkt} \geq 0 , \qquad \forall\ i\in V\backslash\{0\}, p\in P, k\in K, t\in T; \qquad (17)$$

$$u^{B}_{ipkt} \in \{0,1\} , \qquad \forall\ i\in V\backslash\{0\}, p\in P, k\in K, t\in T; \qquad (18)$$

$$y_{pkt} \in \{0,1\} , \qquad \forall\ p\in P, k\in K, t\in T; \qquad (19)$$

$$z_{ikt} \in \{0,1\} , \qquad \forall\ i\in V, k\in K, t\in T; \qquad (20)$$

$$x_{0jkt} \in \{0,1,2\} , \qquad \forall\ j\in V\backslash\{0\}, k\in K, t\in T; \qquad (21)$$

$$x_{ijkt} \in \{0,1\} , \qquad \forall\ i,j\in V\backslash\{0\}, i<j, k\in K, t\in T. \qquad (22)$$

The objective function which minimizes the total cost is represented by expression (1). Constraints (2) correspond to the fill level balance constraints and ensure the correct computation of fill levels between two periods. Constraints (3) limit the amount of glass waste which can be collected in a certain period from a certain container by the amount of glass waste which is stored in this container at the end of the previous period. This ensures that the defined sequence of collection activities and fill level changes is respected. For modeling the requirements of (2) and (3) with respect to the initial fill levels in period 1, constraints (4) and (5) are introduced, respectively. Constraints (6) link the continuous and binary collection variables to each other, and constraints (7) ensure that a container is always emptied completely by a single vehicle. The vehicle capacities are limited by constraints (8), and the maximum number of compartments per vehicle is restricted by constraints (9) and (10). More precisely, constraints (9) ensure that if a certain product type is collected by a certain vehicle, this vehicle also needs to have a compartment assigned to this product type, while constraints (10) limit the number of product types to be transported per vehicle. The routing aspects of the problem are considered in constraints (11) to (15). Constraints (11) ensure that if a container is emptied by a certain vehicle, then the corresponding location must also be visited by this vehicle. Constraints (12) guarantee that the depot is included in each tour while constraints (13) limit the number of vehicles to be used. The node degree constraints and the subtour elimination constraints are represented by equalities (14) and inequalities (15), respectively. Finally, the variable domains are defined in (16) to (22).

3 Literature Review

Several variants of single-period vehicle routing problems with multiple compartments have been discussed in the literature, especially during the last decade. Early papers of this kind have been published by Brown and Graves (1981), Avella et al. (2004), and El Fallahi et al. (2008). Moreover, many papers deal with vehicle routing problems with multiple periods but without considering vehicles with multiple compartments. For surveys on the two main variants of multi-period vehicle routing problems, the periodic vehicle routing problem (PVRP) and the inventory routing problem (IRP), the reader is referred to Campbell and Wilson (2013) for the PVRP and to Bertazzi and Speranza (2012) and Coelho et al. (2014) for the IRP. Only a few publications, however, deal with both problem aspects at the same time, namely a multi-period planning horizon and vehicles with multiple compartments.

The first examination of a multi-period variant of the MCVRP has been published by Cornillier et al. (2008). In their problem, a heterogeneous fleet of multi-compartment vehicles is used to perform the replenishment of petrol products to petrol stations. Each of the stations has a constant consumption rate of different petrol products which are stored in underground tanks. Similar to the problem described in this paper, the main decisions involve the determination of a collection schedule and corresponding vehicle routes. However, the maximal number of locations to be visited on a single tour is limited to two stations, the number and sizes of the compartments are fixed, and each compartment can only store the demand of one petrol station. Moreover, the authors consider a maximal tour duration motivated by the number of working hours per day as well as the option of overtime. In their objective function, they consider revenues generated by the distribution of the petrol products, working time costs and overtime costs, as well as traveling costs. For solving the problem, they propose a multi-phase heuristic. Similar problems are considered by Popović et al. (2012) and Vidović et al. (2014). In their problem variants, homogeneous vehicles with given compartments and compartment sizes are available for the distribution of petrol products to at most three or four petrol stations while each station is only allowed to be visited by one vehicle per period. In addition to traveling costs, Popović et al. (2012) consider inventory holding costs for the petrol stations in the objective function, and they introduce a randomized variable neighborhood descent algorithm as a solution procedure. Vidović et al. (2014) additionally take fleet size costs into account and solve their problem by means of a heuristic approach which combines a matheuristic component for the determination of an initial solution and a variable neighborhood descent algorithm to gain further improvements. Coelho and Laporte (2015) analyze several variants of MCVRPs with multiple periods and heterogeneous vehicles in a petrol replenishment context with inventory holding components. They identify that problems in such a context can usually be distinguished by two characteristics: (1) Compartments can either be used for the distribution of fuel products to only one customer (unsplit compartments) or several customers (split compartments), and (2) a tank containing fuel at a customer site can either be filled by a single vehicle (unsplit tanks) or by multiple vehicles (split tanks) in one period. For the four resulting combinations they propose two types of models each: One model type in which product types are assigned a-priori to compartments (implicit assignment) and one model type in which this assignment is a decision (explicit assignment). Moreover, they adapt and introduce a large number of valid inequalities and symmetry breaking constraints. In order to solve the problem variants, they propose a branch-and-bound algorithm for small problem sizes and a branch-and-cut algorithm for larger problem sizes. According to their classification, the problem described in this paper takes split compartments and unsplit tanks into account.

Lahyani et al. (2015) study a MCVRP-MP for the collection of olive oil of different quality grades from oil producers by a heterogeneous fleet of vehicles with given compartments and compartments sizes. In this problem, the quantities to be collected in each period are explicitly given in advance. The reason of considering several periods simultaneously, nevertheless, is related to cleaning activities which have to be performed to compartments if olive oil of the lowest quality grade has been collected in a period. Such cleaning activities are necessary in order to prevent contamination of olive oils of better qualities in subsequent periods. The objective of this problem is to determine collection schedules which minimize the sum of traveling costs, cleaning costs, and fixed vehicle costs. For the solution of this problem, a branch-and-cut algorithm is presented.

Only recently, Elbek and Wøhlk (2016) dealt with an MP-MCVRP in the context of waste collection. They consider two types of waste, paper and glass waste, which have to be collected from containers in an urban area by a single vehicle which carries two larger containers, i.e. compartments. The collected wastes need to be transported to an intermediate facility at which the vehicle container for paper can be emptied and the vehicle container for glass waste can be exchanged if it is completely filled. If it is not completely filled, the glass waste remains inside the vehicle container, thus, resulting in a reduced capacity for the subsequent period. Eventually, if a certain amount of paper or glass waste has been accumulated at the intermediate facility, it needs to be transported to distinct recycling facilities by the same vehicle. Furthermore, the amount of waste by which a container gets filled in each period is not known in advance. The authors solve their problem by means of a look-ahead heuristic which is implemented in a rolling planning horizon framework.

In contrast to all of the MCVRP-MP variants from the literature, the problem investigated in this paper considers flexible compartment sizes and a flexible number of compartments for the collection vehicles.

4 Matheuristic Algorithm

4.1 Overview

For determining solutions for the MCVRP-MP, we present a heuristic algorithm which iteratively solves a mathematical model in a tabu search framework. A similar approach for the team orienteering arc routing problem has been proposed by Archetti et al. (2015). The motivation for including a mathematical model in the algorithm stems from the idea that a model might be able to efficiently grasp the interdependencies of the many components of the problem, simultaneously. Matheuristics are frequently used to solve routing problems. A recent review on different types of matheuristics and their applications is given in Archetti and Speranza (2014). The ideas of tabu search have been introduced by Glover (1986).

An overview of the matheuristic for the MCVRP-MP is given in Figure 1: In the beginning, an initial solution and, thus, a first incumbent solution is determined by means of a *construction procedure* and a *route improvement procedure*. In each iteration of the subsequent *tabu search phase*, a *mixed integer* *program* (MIP) is solved in order to find a best neighbor to the incumbent solution in a very large neighborhood structure. The newly determined solution is *evaluated* and becomes the new incumbent solution. Furthermore, a *tabu status* is implied on some changes performed. This restricts the set of changes in subsequent iterations. To further diversify the search process, a *jump procedure* is triggered if the algorithm loops around a local optimum. Finally, the algorithm stops when a predefined *termination criterion* is satisfied and the best solution found by the algorithm becomes the solution of the algorithm. The mentioned components of the procedure are explained in more detail in the following.

```
input: problem data;

x_0^best = ∞;
generate initial solution x;
apply route improvement procedure to x;
x^best := x;
do
   do
      solve MIP to determine best neighbor x' of the incumbent solution x;
      update tabus;
      apply route improvement procedure to x';
      x := x';
      if x_0(x) ≤ x_0^best then
         x_0^best := x_0(x);
         x^best := x;
      endif
   until jump procedure is triggered;
   perform jump procedure to determine solution x'';
   x := x'';
until termination criterion is satisfied;

output: x^best.
```

Figure 1: Overview of the matheuristic

4.2 Construction Procedure

An initial solution is determined by an extended version of the construction procedure described by Henke et al. (2015). In a first step, the multi-period planning horizon is separated into single periods. This separation is realized by simulating the fill levels of all containers over time and by assuming that each container is emptied as late as possible, i.e. in the last period before its capacity would be exceeded. By means of this simulation, for each period the supply to be collected from each container is obtained.

In a second step, for each period an assignment of supplies to vehicles is determined. For a single period, a random sequence of all positive supplies is generated. According to this sequence, starting with vehicle $k = 1$ each supply is then assigned to a vehicle with respect to the following conditions: A supply can only be assigned to a vehicle if (1) the corresponding product type is already assigned to this vehicle or if it can be additionally assigned to this vehicle, and (2) if the vehicle capacity would not be exceeded by adding this supply. If it is not possible to assign the supply to the regarded vehicle k, then the next vehicle $k + 1$ is checked. This procedure is continued until the considered supply can eventually be assigned to a vehicle. If the available number of vehicles is not sufficient, auxiliary vehicles are created.

In a third step, for each period and for each vehicle, a sequence according to which the locations are visited, i.e. a route, is determined. In order to do so, each route is simply built according to the sequence in which locations have been added to a vehicle in step 2.

4.3 Route Improvement Procedure

After an initial solution is determined and each time after the MIP has found a new solution, all routes in the incumbent solution are improved by applying the well-known Lin-Kernighan-Heuristic (Lin and Kernighan, 1973). For this, the code of Helsgaun (2000) has been implemented.

4.4 Tabu Search Phase

4.4.1 Neighborhood Structure

In each iteration of the algorithm, a MIP is solved for the determination of the best non-tabu neighbor from a very large neighborhood structure of the incumbent solution. The underlying operator allows for moving from one solution to another by changing an unspecified number of supply assignments between periods and vehicles, i.e. an unspecified number of supplies can be moved from their current tour to others. Such movements of supplies might consequently lead to changes of location assignments, i.e. locations could be eliminated from and/or added to tours. In contrast to changes of supply assignments, the number of changes of location assignments is limited for each tour, e.g. at most three locations can be eliminated from or added to a tour. The MIP represents the elimination and addition of supplies and locations from and to tours as decision variables, and further respects all aspects of the planning problem described in Section 2, except for the routing components. Instead of modeling the routing components exactly, approximated savings and insertion costs for each container location are determined a-priori and are considered as coefficients in the objective function. This will be described in more detail in the following.

4.4.2 Solution Evaluation

With respect to the evaluation of a solution, two different objective functions are used. The first objective function consists of the total costs, as defined in (1), and a penalty term for auxiliary vehicles. This objective function is used whenever a new solution is obtained from the construction procedure, from the route improvement procedure or from the tabu search phase.

The second objective function is used in the model for the evaluation of neighbors during the neighborhood search and considers savings and insertion costs which occur from eliminating locations from or adding locations to tours, respectively, as well as the penalty term for auxiliary vehicles. The savings and insertion costs are determined before the model is solved in the following manner: For each combination of a period, a vehicle, and a location in the incumbent solution it is checked whether the considered location is visited on the considered tour. If it is visited, the savings (sav_{ikt}) due to the elimination of this location from tour k in period t, in which it is currently visited between locations h and j, are calculated by equation (23), assuming $h < i < j$ for simplicity. If the location is not visited, the cost ($cost_{ikt}$) of adding this location in all possible positions of the tour are calculated according to equation (24). Consequently, only the position with the lowest increase in cost is further considered.

$$sav_{ikt} = c_{hi} + c_{ij} - c_{hj} ; \qquad (23)$$
$$cost_{ikt} = c_{hi} + c_{ij} - c_{hj} . \qquad (24)$$

Savings and insertion costs are only guaranteed to be exact if at most one location is added to or eliminated from a tour. If more than one location is changed, then savings and insertion costs represent approximations of the change in the total cost. These approximations may become more inaccurate with an increasing number of changes of location assignments per tour. Therefore, the number of such changes per tour is limited. In addition, pre-tests have shown that a random manipulation of the savings improves the solution quality of the algorithm. Therefore, all savings which are used in the objective function are increased randomly in the interval $[sav_{ikt}; (1 + \varepsilon) \cdot sav_{ikt}]$, $\varepsilon \geq 0$.

The penalty term, which is used in both objective functions, considers the auxiliary vehicles which might have been introduced in the construction phase. For each auxiliary vehicle, a high penalty factor is added to the objective functions with the intention of guiding the algorithm to solutions with fewer and eventually no auxiliary vehicles. Moreover, each time a solution with fewer auxiliary vehicles is found, the number of auxiliary vehicles cannot be increased again.

4.4.3 Tabu Statuses and Aspiration Criterion

Each time after a new incumbent solution has been determined, all location changes, which have been performed to move from the previous solution to the new solution, are set tabu for being reversed for a fixed number of iterations. However, all tabus can be ignored if this would allow us to find a new best solution since the last jump. As this aspiration criterion should only be considered if the resulting difference between savings and insertion costs corresponds to the actual change in total cost, it can only be activated if at most one location is eliminated from or added to each tour.

4.4.4 Model

In order to model the described neighborhood operations, the following set, variables and constants are introduced in addition to the symbols of model (1) – (22):

Set:

$\overline{K}$: set of available and auxiliary vehicles.

Variables:

$$\delta^{s-}_{ipkt} = \begin{cases} 1, \text{ if supply (i,p) is eliminated from vehicle k in period t,} \\ 0, \text{ otherwise,} \end{cases} \quad \forall\ i \in V \backslash \{0\},\ p \in P,\ k \in \overline{K},\ t \in T;$$

$$\delta^{s+}_{ipkt} = \begin{cases} 1, \text{ if supply (i,p) is added to vehicle k in period t,} \\ 0, \text{ otherwise,} \end{cases} \quad \forall\ i \in V \backslash \{0\},\ p \in P,\ k \in \overline{K},\ t \in T;$$

$$\delta^{l-}_{ikt} = \begin{cases} 1, \text{ if location i is eliminated from vehicle k in period t,} \\ 0, \text{ otherwise,} \end{cases} \quad \forall\ i \in V \backslash \{0\},\ k \in \overline{K},\ t \in T;$$

$$\delta^{l+}_{ikt} = \begin{cases} 1, \text{ if location i is added to vehicle k in period t,} \\ 0, \text{ otherwise,} \end{cases} \quad \forall\ i \in V \backslash \{0\},\ k \in \overline{K},\ t \in T;$$

$T = \begin{cases} 1, \text{ if the aspiration criterion is triggered,} \\ 0, \text{ otherwise;} \end{cases}$

$\bar{z}_{kt} = \begin{cases} 1, \text{ vehicle k is used in period t,} \\ 0, \text{ otherwise,} \end{cases}$ $\quad \forall\ k\in\bar{K},\ t\in T;$

K^{aux} : number of auxiliary vehicles needed;

δ_0 : net savings (objective function value).

Constants:

ρ : penalty factor;

$\hat{u}^B_{ipkt} = \begin{cases} 1, \text{ if supply (i,p) is collected by vehicle k in period t} \\ \quad \text{in the incumbent solution,} \\ 0, \text{ otherwise,} \end{cases}$ $\quad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\bar{K},\ t\in T;$

Δ^{max} : Maximal number of locations to be eliminated from a single vehicle;

Δ^{min} : Minimal number of supplies to be eliminated from or added to any vehicle;

$\tau^-_{ipkt} = \begin{cases} 1, \text{ if the elimination of supply (i,p) from vehicle k} \\ \quad \text{in period t is tabu,} \\ 0, \text{ otherwise,} \end{cases}$ $\quad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\bar{K},\ t\in T;$

$\tau^+_{ipkt} = \begin{cases} 1, \text{ if the addition of supply (i,p) to vehicle k in period t} \\ \quad \text{is tabu,} \\ 0, \text{ otherwise,} \end{cases}$ $\quad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\bar{K},\ t\in T;$

x_0^{inc} : total cost of the incumbent solution;

x_0^{best} : total cost of the best solution found since the last jump;

M : sufficiently large number.

The model can then be formulated as follows:

$$\max \delta_0 = \sum_{\substack{i\in V \\ i>0}} \sum_{k\in\bar{K}} \sum_{t\in T} (sav_{ikt} \cdot \delta^{l-}_{ikt} - cost_{ikt} \cdot \delta^{l+}_{ikt}) - \rho \cdot K^{aux} \ ; \tag{25}$$

$$\delta^{s-}_{ipkt} \leq \hat{u}^B_{ipkt}\ , \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\bar{K},\ t\in T; \tag{26}$$

$$\delta^{s+}_{ipkt} \leq 1 - \hat{u}^B_{ipkt}\ , \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\bar{K},\ t\in T; \tag{27}$$

$$\sum_{p\in P} (\hat{u}^B_{ipkt} + \delta^{s+}_{ipkt} - \delta^{s-}_{ipkt}) \leq \hat{m} \cdot (1 - \delta^{l-}_{ikt})\ , \qquad \forall\ i\in V\backslash\{0\},\ k\in\bar{K},\ t\in T; \tag{28}$$

$$\delta^{l-}_{ikt} \leq \sum_{p\in P} \delta^{s-}_{ipkt}\ , \qquad \forall\ i\in V\backslash\{0\},\ k\in\bar{K},\ t\in T; \tag{29}$$

$$\sum_{p\in P} \hat{u}^B_{ipkt} - |P| \cdot \sum_{p\in P} \left(\hat{u}^B_{ipkt} + \delta^{s+}_{ipkt} - \delta^{s-}_{ipkt}\right) \leq \hat{m} \cdot \delta^{l-}_{ikt}\ , \qquad \forall\ i\in V\backslash\{0\},\ k\in\bar{K},\ t\in T; \tag{30}$$

$$\sum_{p\in P} \delta^{s+}_{ipkt} - |P| \cdot \sum_{p\in P} \hat{u}^B_{ipkt} \leq \hat{m} \cdot \delta^{l+}_{ikt}\ , \qquad \forall\ i\in V\backslash\{0\},\ k\in\bar{K},\ t\in T; \tag{31}$$

$$\delta^{l+}_{ikt} \leq \sum_{p\in P} \delta^{s+}_{ipkt}\ , \qquad \forall\ i\in V\backslash\{0\},\ k\in\bar{K},\ t\in T; \tag{32}$$

$$\sum_{p\in P} \hat{u}^B_{ipkt} \leq \hat{m} \cdot (1 - \delta^{l+}_{ikt})\ , \qquad \forall\ i\in V\backslash\{0\},\ k\in\bar{K},\ t\in T; \tag{33}$$

$$\sum_{\substack{i\in V\\i>0}}\sum_{k\in K}\sum_{t\in T}(\delta^{l-}_{ikt}+\delta^{l+}_{ikt})\geq\Delta^{min}\,; \qquad (34)$$

$$\sum_{\substack{i\in V\\i>0}}(\delta^{l-}_{ikt}+\delta^{l+}_{ikt})\leq\Delta^{max}\,, \qquad \forall\ k\in\overline{K},\ t\in T; \qquad (35)$$

$$l_{ipt}=l_{ip,t-1}+f_{ip}-\sum_{k\in\overline{K}}u^{R}_{ipkt}\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ t\in T\backslash\{1\}; \qquad (36)$$

$$\sum_{k\in\overline{K}}u^{R}_{ipkt}\leq l_{ip,t-1}\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ t\in T\backslash\{1\}; \qquad (37)$$

$$l_{ip1}=l^{init}_{ip}+f_{ip}-\sum_{k\in\overline{K}}u^{R}_{ipkt}\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P; \qquad (38)$$

$$\sum_{k\in\overline{K}}u^{R}_{ipk1}\leq l^{init}_{ip}\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P; \qquad (39)$$

$$u^{R}_{ipkt}\leq q_{ip}\cdot(\hat{u}^{B}_{ipkt}+\delta^{s+}_{ipkt}-\delta^{s-}_{ipkt})\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\overline{K},\ t\in T; \qquad (40)$$

$$l_{ipt}\leq f_{ip}+(q_{ip}-f_{ip})\cdot\left(1-\sum_{k\in\overline{K}}(\hat{u}^{B}_{ipkt}+\delta^{s+}_{ipkt}-\delta^{s-}_{ipkt})\right), \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ t\in T; \qquad (41)$$

$$\sum_{i\in V\backslash\{0\}}\sum_{p\in P}u^{R}_{ipkt}\leq Q\,, \qquad \forall\ k\in\overline{K},\ t\in T; \qquad (42)$$

$$\hat{u}^{B}_{ipkt}+\delta^{s+}_{ipkt}-\delta^{s-}_{ipkt}\leq y_{pkt}\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\overline{K},\ t\in T; \qquad (43)$$

$$\sum_{p\in P}y_{pkt}\leq\hat{m}\cdot\overline{z}_{kt}\,, \qquad \forall\ k\in\overline{K},\ t\in T; \qquad (44)$$

$$\sum_{k\in\overline{K}}\overline{z}_{kt}-|K|\leq K^{aux}\,, \qquad \forall\ t\in T; \qquad (45)$$

$$\overline{z}_{k+1,t}\leq\overline{z}_{kt}\,, \qquad \forall\ k\in\overline{K}\backslash\{|\overline{K}|\},\ t\in T; \qquad (46)$$

$$\delta^{l-}_{ikt}\leq 1-\tau^{-}_{ikt}+T\,, \qquad \forall\ i\in V\backslash\{0\},\ k\in\overline{K},\ t\in T; \qquad (47)$$

$$\delta^{l+}_{ikt}\leq 1-\tau^{+}_{ikt}+T\,, \qquad \forall\ i\in V\backslash\{0\},\ k\in\overline{K},\ t\in T; \qquad (48)$$

$$x^{inc}_{0}-\delta_{0}-x^{best}_{0}\leq M\cdot(1-T)\,, \qquad (49)$$

$$\sum_{\substack{i\in V\\i>0}}(\delta^{l-}_{ikt}+\delta^{l+}_{ikt})\leq 1+|V|\cdot(1-T), \qquad \forall\ k\in\overline{K},\ t\in T; \qquad (50)$$

$$l_{ipt}\geq 0\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ t\in T; \qquad (51)$$

$$u^{R}_{ipkt}\geq 0\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\overline{K},\ t\in T; \qquad (52)$$

$$y_{pkt}\in\{0,1\}\,, \qquad \forall\ p\in P,\ k\in\overline{K},\ t\in T; \qquad (53)$$

$$\overline{z}_{kt}\in\{0,1\}\,, \qquad \forall\ k\in\overline{K},\ t\in T; \qquad (54)$$

$$K^{aux}\geq 0\,; \qquad (55)$$

$$\delta^{s-}_{ipkt}\in\{0,1\}\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\overline{K},\ t\in T; \qquad (56)$$

$$\delta^{s+}_{ipkt}\in\{0,1\}\,, \qquad \forall\ i\in V\backslash\{0\},\ p\in P,\ k\in\overline{K},\ t\in T; \qquad (57)$$

$$\delta^{l-}_{ikt}\in\{0,1\}\,, \qquad \forall\ i\in V\backslash\{0\},\ k\in\overline{K},\ t\in T; \qquad (58)$$

$$\delta^{l+}_{ikt} \in \{0,1\}, \qquad \forall\ i \in V \backslash \{0\}, k \in \overline{K}, t \in T; \tag{59}$$

$$\mathrm{T} \in \{0,1\}. \tag{60}$$

Expression (25) represents the second objective function described in Section 4.4.2. Constraints (26) ensure that a supply can only be eliminated from a tour if it is assigned to this tour in the incumbent solution whereas constraints (27) work analogously for the addition of a supply to a tour. Constraints (28) – (30) guarantee that a location can only be eliminated from a tour if all supplies of this location are eliminated from this tour. In a similar manner, constraints (31) – (33) guarantee that a location can only be added to a tour if at least one supply of this location is added to the considered vehicle. In order to ensure that at least one location assignment is changed, constraints (34) are introduced, while constraints (35) limit the number of changes of location assignments for each tour. Constraints (36) – (41) are similar to constraints (2) – (7) and they model the development of the fill levels over time. Constraints (42) – (44) limit the vehicle capacities and the maximum number of compartments per vehicle. The number of vehicles used in the solution is determined by constraints (45) whereas constraints (46) enforce that vehicles are used in such a way that a vehicle i can only be used for collection if vehicle i - 1 is also used. Constraints (47) and (48) take the tabu statuses into account whereas constraints (49) and (50) model the aspiration criterion. Finally, constraints (51) to (60) define the variable domains.

4.4.5 Jump Procedure

Despite the use of tabus, the algorithm sometimes loops around similar solutions after a local optimum has been found. In order to overcome this problem, a diversification procedure, also called jump procedure, is included in the algorithm. The main idea of this procedure is to enforce a considerable change of the incumbent solution. In the algorithm, this is implemented by slightly changing and solving the model (25) – (60). Instead of limiting the number of changes of location assignments per tour, a minimal number of such changes is enforced by replacing constraints (35) with constraints (61), i.e. a minimal number of locations must be eliminated from each tour. For each tour, this minimal number is determined as a specified percentage γ of the currently visited locations, e.g. 50 percent of the currently visited locations and the corresponding supplies must be eliminated from each tour. Furthermore, all tabu statuses are ignored and newly initialized after a jump was performed.

$$\sum_{\substack{i \in V \\ i>0}} \delta^{l-}_{ikt} \geq \gamma \cdot \sum_{\substack{i \in V \\ i>0}} \overline{u}^{B}_{ikt}, \qquad \text{with } \overline{u}^{B}_{ikt} = \begin{cases} 1, \text{ if } \sum_{p \in P} \hat{u}^{B}_{ipkt} \geq 1, \\ 0, \text{ else,} \end{cases} \qquad \forall\ k \in \overline{K}, t \in T. \tag{61}$$

The jump procedure is applied to the incumbent solution as soon as the best solution found since the last jump or since the start of the algorithm (in case no jump was performed, yet) has not been improved for a certain number of iterations.

4.4.6 Termination Criterion

The algorithm either terminates after a certain number of iterations have been performed or after a certain time limit is reached.

5 Numerical Experiments

5.1 Overview

In order to gain detailed insights into the introduced planning problem and into the performance of the matheuristic, three sets of experiments have been conducted. The research questions behind these experiments are as follows and will be investigated in the following:

(1) What is the impact of the different problem parameters on determining feasible or even optimal solutions of the MCVRP-MP?
(2) What is the performance of the matheuristic with respect to its solution quality and computing time?
(3) What is the benefit of planning for several periods simultaneously instead of considering single periods, only?

Due to the novelty of the problem variant, no benchmark instances had been available in order to conduct the experiments and, therefore, new instances were generated. The characteristics of these instances and the procedure to obtain the instances will be explained in Section 5.2, before the details of the three sets of experiments are described and their results are discussed in Sections 5.3 – 5.5.

All algorithms used in the experiments have been implemented in C++ with an interface to CPLEX 12.6. Computers with 3.2 gigahertz and 8 gigabytes RAM have been used for all experiments.

5.2 Problem Instances

As controllable parameters for the newly generated instances, the number of locations, the number of product types, the maximal number of compartments, the number of periods, and a factor $\bar{q}$ which controls the vehicle capacity have been selected. For a given combination of these parameters an instance is generated in the following manner: In each instance, the depot is located in the center of a $100 \cdot 100$ length units area, and for each location, coordinates with two digits are selected randomly from the interval [0, 100]. The vehicle capacity is determined by $30 \cdot |V| \cdot |P| \cdot \bar{q}$ and the container capacities are set to 100 for each container. This combination of capacities results in instances, in which about three vehicles are used if the capacity factor is set equal to one and in instances with about six vehicles if the capacity factor is set to 0.5. Furthermore, for each container (i, p), a fill rate is selected randomly from the interval [1, q_{ip}] and the initial fill level is determined randomly out of the set of all integer multiples of the corresponding fill rate which ensure that the container has to be emptied at least once during the planning horizon. Finally, the number of available vehicles is determined by solving a modified bin packing problem with CPLEX. In this problem, the initial fill levels of all containers have to be assigned to vehicles in a single period such that the vehicle capacities and the maximum number of compartments are respected and the number of necessary vehicles is minimized.

For the experiments, the following specifications have been chosen for the adjustable problem parameters:

- Number of locations $|V| \in \{10, 20, 30, 40, 50\}$;
- Number of product types $|P| \in \{3, 4\}$;
- Maximum number of compartments $\widehat{\mathrm{m}} \in \{2, 3\}$ for $|P| = 3$ and $\widehat{\mathrm{m}} \in \{2, 3, 4\}$ for $|P| = 4$;
- Number of periods $|T| \in \{3, 6\}$;
- Vehicle capacity factor $\overline{\mathrm{q}} \in \{0.5, 1\}$.

For each of the resulting 100 combinations of these parameters, a single instance has been generated randomly. All instances can be downloaded from http://www.mansci.ovgu.de/mansci/en/Research/Materials.html.

5.3 Insights into the Complexity of the Problem

As the introduced variant of the MCVRP-MP has not been investigated in the literature before, the impact of different problem characteristics on determining feasible and optimal solutions to the problem has been examined in the first set of experiments. The model introduced in Section 2, which represents all components of the planning problem, has been implemented into CPLEX by adding subtour elimination constraints through a separation procedure, i.e. in the beginning, no subtour elimination constraints are considered, and then, in each integral node found by the solver, a subtour elimination constraint is added to the pool of constraints which prohibits the occurrence of the subtour with the smallest number of nodes. Furthermore, symmetry breaking constraints have been added which ensure that for each period, the total cost for each vehicle is non-increasing with the vehicle indices. Each of the 100 instances has been solved with a time limit of one hour and the resulting total cost as well as the gaps have been recorded.

$\|V\|$	**#inst**	**#feas**	**rel.feas**	**#opt**	**rel.opt**	**avg.gap**
10	20	17	85%	3	15%	14.6%
20	20	11	55%	0	0%	20.9%
30	20	5	25%	0	0%	19.3%
40	20	4	20%	0	0%	17.7%
50	20	0	0%	0	0%	n.a.

Table 1: Feasible and optimal solutions determined by the exact approach, clustered according to the number of locations

Table 1 summarizes the results clustered according to the number of locations. The columns in this table depict, for each cluster, the number of instances (#inst), the absolute and relative number of instances for which a feasible solution has been determined (#feas, rel.feas), the absolute and relative number of instances for which an optimal solution has been determined (#opt, rel. opt), and the average gap (avg.gap).

The results demonstrate that with an increasing number of locations, the number of feasibly solved instances decreases from 85% for instances with 10 locations to 0% for instances with 50 locations. Moreover, the exact method was only able to find three optimal solutions for all instances within the given time limit, all of which belong to the cluster in which 10 locations are considered. These findings show that the problem is very difficult to solve to optimality with a simple exact method, and the results furthermore suggest that the problem becomes significantly more difficult to solve with an increasing number of locations.

Tables 2 – 5 show similar results for the remaining four parameters. From these results, it may be deduced that the difficulty of solving the MCVRP-MP increases with an increasing number of product types, a decreasing maximal number of compartments, an increasing number of periods, and a decreasing vehicle capacity. The effects w.r.t. the number of product types and periods correspond to the expected outcome as these parameters have a direct impact on the size of the model. As a decreasing vehicle capacity increases the number of necessary vehicles, also this finding is consistent to the expected outcome. The results w.r.t. to maximum number of compartments can be explained as follows: When the maximal number of compartments is small in relation to the number of product types, the decision of assigning product types to vehicles becomes harder, and this decision seems to have a significant impact on the problem difficulty. Except for the vehicle capacity, similar impacts of the considered parameters on the solvability of a single-period problem variant of the MCVRP were reported by Henke et al. (2015).

$\|P\|$	#inst	#feas	rel.feas	#opt	rel.opt	avg.gap
3	40	16	40%	1	6%	16.2%
4	60	21	35%	2	10%	18.4%

Table 2: Feasible and optimal solutions determined by the exact approach, clustered according to the number of product types

$\|P\|$	$\hat{m}$	#inst	#feas	rel.feas	#opt	rel.opt	avg.gap
3	2	20	7	35%	0	0%	16.9%
3	3	20	9	45%	1	11%	15.7%
4	2	20	7	35%	1	14%	22.3%
4	3	20	6	30%	0	0%	25.7%
4	4	20	8	40%	1	13%	9.5%

Table 3: Feasible and optimal solutions determined by the exact approach, clustered according to the maximum number of compartments

Interestingly, the vehicle capacity seems to have a stronger impact on the difficulty of the problem than the number of periods does. Although increasing the number of periods from 3 to 6 has a stronger impact on the model size than decreasing the vehicle capacity from 1 to 0.5, the difference in feasibly solved instances does not reflect this increase in problem size. It can, therefore, be deduced that in the MCVRP-MP, the assignment of supplies to vehicles is a more difficult decision than the assignment of supplies to periods.

$\|T\|$	#inst	#feas	rel.feas	#opt	rel.opt	avg.gap
3	50	25	50%	3	12%	17.2%
6	50	12	24%	0	0%	18.0%

Table 4: Feasible and optimal solutions determined by the exact approach, clustered according to the number of periods

$\bar{q}$	#inst	#feas	rel.feas	#opt	rel.opt	avg.gap
0.5	50	8	16%	0	0%	26.5%
1	50	29	58%	3	10%	15.0%

Table 5: Feasible and optimal solutions determined by the exact approach, clustered according to the vehicle capacity factor

5.4 Performance of the Matheuristic

In order to investigate the performance of the matheuristic, each instance was solved five times with a computing time limit set to 20 minutes. Furthermore, the best solution found over these runs has been used as a starting solution for the exact solution approach which was run for another hour. In preliminary experiments, the best settings for the algorithm parameters have been determined as follows:

- number of iterations for which a performed change of a location assignment cannot be reversed within the tabu search phase: 5;
- maximal number of changes of location assignments per vehicle in one iteration of the tabu search phase: 6;
- number of iterations without improvement of the best solution found since the last jump after which a new jump is triggered: 15;
- percentage of the locations currently assigned to a vehicle which have to be eliminated from this vehicle in a jump (γ): 50%;
- factor for determining randomized savings (ε): 0.1.

Table 6 presents the results for all instances clustered according to the number of locations. In columns 3 to 5, results from the five runs of the matheuristic are shown. More precisely for each cluster, these columns list the average of the best objective function values found in the five runs of the matheuristic (avg.bhs, average of the best heuristic solutions), the average deviation from these best objective function values (avg.dev1), and the maximal deviation from these best objective function values (max.dev). Furthermore in columns 6 to 8, results for the comparison of the matheuristic with the results of the exact approach are given. The objective function values considered in these columns correspond to the solutions found by the exact approach which, for each instance, was started from the best heuristic solution. For each cluster, the columns give the average objective function values of the best known solutions (avg.bks), the number of instances for which the exact solution approach found better solutions than the matheuristic (#imp), the average relative improvement gained by the exact approach (avg.imp), and the average deviation of the objective functions values of the heuristic solutions from the best known solutions (avg.dev2).

|V|	**#inst**	**only matheuristic solutions considered**			**exact and matheuristic solutions considered**			
		avg.bhs	**avg.dev1**	**max.dev**	**avg.bks**	**#imp**	**avg.imp**	**avg.dev2**
10	20	2135.2	0.25%	2.36%	2128.5	4	0.09%	0.34%
20	20	2748.8	0.59%	3.43%	2730.9	7	0.12%	0.72%
30	20	3300.8	1.16%	4.95%	3260.9	3	0.01%	1.17%
40	20	3693.1	1.32%	5.31%	3646.2	5	0.06%	1.39%
50	20	4128.7	1.73%	15.16%	4046.5	5	0.01%	1.74%

Table 6: Quality of the solutions obtained by the matheuristic, clustered according to the number of locations

Table 6 demonstrates that the average deviations of the objective function values from the best heuristic solutions are below 2% for each cluster. For small instances with only 10 locations, the average deviation amounts to only 0.25% whereas for larger instances with 50 locations this deviation increases to 1.73%.

With respect to the exact solutions, it was possible to improve between 3 and 7 instances in each cluster, where the improvements gained by the exact approach are relatively small with 0.01% to 0.12%, on average. The number of locations does not seem to have a clear impact on the gained improvements; however, a small trend towards smaller improvements can be observed.

|P|	**#inst**	**only matheuristic solutions considered**			**exact and matheuristic solutions considered**			
		avg.bhs	**avg.dev1**	**max.dev**	**avg.bks**	**#imp**	**avg.imp**	**avg.dev2**
3	40	2940.4	0.86%	4.95%	2912.4	10	0.06%	0.93%
4	60	3375.3	1.11%	15.16%	3329.3	14	0.05%	1.17%

Table 7: Quality of the solutions obtained by the matheuristic, clustered according to the number of product types

Tables 7 – 10 give similar results with respect to the four remaining parameters. From these tables, it can be concluded that the performance of the matheuristic decreases with an increasing number of product types, a decreasing number of maximal compartments, and an increasing number of periods. In contrast to the previous findings with respect to the vehicle capacity, an effect contrary to the exact approach can be observed for the matheuristic. For the matheuristic, the performance increases with a decreasing capacity, and, thus, with an increasing number of vehicles. This finding is further strengthened by the maximal deviations which also decrease with a decreasing capacity. A possible explanation for this finding is that with a lower capacity, fewer locations have to be visited by a vehicle on average. Not surprisingly, it seems as if the matheuristic is better at making reasonable assignment decisions than reasonable routing decisions.

$\|P\|$	$\hat{m}$	#inst	only matheuristic solutions considered			exact and matheuristic solutions considered			
			avg.bhs	avg.dev1	max.dev	avg.bks	#imp	avg.imp	avg.dev2
3	2	20	3502.1	0.89%	3.92%	3468.2	6	0.07%	0.96%
3	3	20	2378.8	0.84%	4.95%	2356.7	4	0.05%	0.89%
4	2	20	4197.3	1.15%	15.16%	4132.1	3	0.01%	1.15%
4	3	20	3509.5	1.27%	9.64%	3459.8	7	0.09%	1.36%
4	4	20	2419.0	0.92%	4.91%	2396.2	4	0.07%	0.99%

Table 8: Quality of the solutions obtained by the matheuristic, clustered according to the maximum number of compartments

$\|T\|$	#inst	only matheuristic solutions considered			exact and matheuristic solutions considered			
		avg.bhs	avg.dev1	max.dev	avg.bks	#imp	avg.imp	avg.dev2
3	50	2097.6	0.94%	5.31%	2075.6	11	0.04%	0.98%
6	50	4305.1	1.09%	15.16%	4249.6	13	0.07%	1.16%

Table 9: Quality of the solutions obtained by the matheuristic, clustered according to the number of periods

$\bar{q}$	#inst	only matheuristic solutions considered			exact and matheuristic solutions considered			
		avg.bhs	avg.dev1	max.dev	avg.bks	#imp	avg.imp	avg.dev2
0.5	50	2135.2	0.25%	2.36%	2128.5	4	0.09%	0.34%
1	50	2748.8	0.59%	3.43%	2730.9	7	0.12%	0.72%

Table 10: Quality of the solutions obtained by the matheuristic, clustered according to the vehicle capacity factor

Also with respect to the improvements gained by further trials for improving the best heuristic solutions with the exact approach, the results do not entirely correspond to the previous findings. Whereas for the number of product types, the maximal number of compartments, and the vehicle capacity the results are similar, they differ with respect to the number of periods. The exact approach was able to improve slightly more solutions for the instances with six periods than for the instances with three periods.

5.5 Benefits of Multi-Period Planning

In order to generate managerial insights with respect to the benefits of multi-period planning, additional analyses have been performed. Single-period planning in general reduces flexibility as a rule for determining an a-priori collection schedule must be defined, i.e. the assignment of supplies to periods is no longer a decision type in the planning problem. When multiple periods are considered simultaneously, savings in total cost might be obtained because the determination of a collection schedule does no longer only depend on the fill rates and container capacities, but also on the traveling costs. The following experiments have been designed with the objective of determining how large the benefits of multi-period planning are.

For the experiments, the best known solutions for all multi-period instances with 10 and 20 locations have been taken into account. In order to obtain single-period instances, the filling of containers over the planning horizon has been simulated in such a way that each container is always emptied during the last period before its capacity would be exceeded. Through this simulation, an assignment of supplies to periods and, thus, single-period instances have been derived. These single-period instances have been solved to optimality by the branch-and-cut approach developed by Henke et al. (2017). Finally, the sum of the total costs for all single-period instances which were derived from the same multi-period instance has been compared to the best known total cost determined for the multi-period instance. As the solutions from the matheuristic and the subsequently applied exact approach are not necessarily optimal, the reported differences between single-period planning and multi-period planning should be considered as lower bounds on the benefits of multi-period planning.

Table 11 shows the corresponding results clustered according to the number of locations. For each cluster, the average sum of total cost for the single-period instances (sp avg.cost), the average total cost for the multi-period instances (mp avg.cost) and the relative savings obtainable from multi-period planning (sav) are reported.

$\|V\|$	#inst	sp avg.cost	mp avg.cost	sav
10	20	2256.0	2126.5	5.8%
20	20	2824.3	2727.4	3.3%

Table 11: Savings obtained by multi-period planning clustered according to the number of locations

It can be observed that planning for multiple periods results in average savings of 5.8% for instances with 10 locations, and 3.3% for instances with 20 locations, respectively. On the one hand, this significant change in savings might be explained by the fact that for instances with 20 locations, the average number of locations to be visited per tour becomes larger because of the procedure by which the instances have been generated. A tour with more locations to be visited is assumed to cover a larger part of the geographical area than a tour with fewer locations, and, therefore, the elimination or addition of locations resulting from changes of location assignments between periods might have a lower impact on the total tour length. On the other hand, it has to be noted that the total costs for the multi-period instances are only upper bounds because these instances are not necessarily solved to optimality. As shown in the previous subsection, the performance of the matheuristic decreases with an increasing number of locations and, thus, the difference of the optimal solutions to the obtained upper bounds might increase with an increasing number of locations.

Regarding the four additional parameters the results are as follows: The potential of cost savings by means of multi-period planning increases with a decreasing number of product types (5.0% for three product types and 4.3% for four product types), an increasing number of periods (4.5% for three periods and 4.7% for six periods), and a decreasing vehicle capacity (5.1% for a capacity factor of 0.5 and 4.0%

for a capacity factor of 1.0). For the maximum number of compartments, no clear trend can be observed. Interestingly, the increase in savings which can be obtained by extending the planning horizon from three to six periods is only 0.2%, whereas savings of 4.5% can be obtained by extending the planning horizon from one to three periods. This suggests that a simultaneous consideration of only a small number of periods can already have a large impact on the total cost of a collection schedule.

To sum up, although the reported savings must be considered as lower bounds on the exact savings, there is a significant potential to save costs by simply extending a single-period planning horizon to multiple periods. For the considered instances, the average savings amount to 4.6% of the total cost in single-period planning.

6 Conclusions

In this paper, a new variant of a multi-compartment vehicle routing problem with multiple periods has been introduced which can be found in the context of glass waste collection. A special characteristic of this problem compared to problem variants from the literature is the explicit consideration of flexible compartment sizes. In order to solve the presented problem variant, an exact approach and a matheuristic algorithm have been developed and tested.

The results with respect to the exact approach have shown that the planning problem is very difficult to solve to optimality as only for 3% of the considered instances optimal solutions have been found within one hour. Moreover, only for 37% of the instances feasible solutions have been determined at all.

The developed matheuristic has been shown to solve MCVRP-MP instances with a good solution quality and within acceptable computing times. For all instances, feasible solutions have been found, and, on average, the solutions obtained by the matheuristic deviate by 1.07% from the benchmark solutions.

Finally, experiments with respect to the benefits of considering multiple periods simultaneously instead of single-period planning have shown a significant savings potential: Only by extending the planning horizon from one to three periods, savings in total costs of, on average, 4.5% have been obtained.

One point of criticism on the considered planning problem might be the practicability of the obtained solution as in some real-world applications supplies or demands are not known in advance. Therefore, considering stochasticity would be an interesting feature for further research.

References

Archetti, C.; Speranza, M.G. (2014): A survey on matheuristics for routing problems. In: EURO Journal on Computational Optimization 2, 223-246.

Archetti, C.; Corberán, Á.; Plana, I.; Sanchis, J.M.; Speranza, M.G. (2015): A matheuristic for the team orienteering arc routing problem. In: European Journal of Operational Research 245, 392-401.

Avella, P.; Boccia, M.; Sforza, A. (2004): Solving a fuel delivery problem by heuristic and exact approaches. In: European Journal of Operational Research 152, 170-179.

Bertazzi, L.; Speranza, M.G. (2012): Inventory routing problems: an introduction. In: EURO Journal on Transportation and Logistics 1, 307-326.

Brown, G.G.; Graves, G.W. (1981): Real-time dispatch of petroleum tank trucks. In: Management Science 27, 19-32.

Campbell, A.M.; Wilson, J.H. (2013): Forty years of periodic vehicle routing. In: Networks 63, 2-15.

Coelho, L.C.; Cordeau, J.-F.; Laporte, G. (2014): Thirty years of inventory routing. In: Transportation Science 48, 1-19.

Coelho, L.C.; Laporte, G. (2015): Classification, models and exact algorithms for multi-compartment delivery problems. In: European Journal of Operational Research 242, 854-864.

Cornillier, F.; Boctor, F.F.; Laporte, G.; Renaud, J. (2008): A heuristic for the multi-period petrol station replenishment problem. In: European Journal of Operational Research 191, 295-305.

Dantzig, G.B.; Ramser, J.H. (1959): The truck dispatching problem. In: Management Science 6, 80-91.

El Fallahi, A.; Prins, C.; Wolfer Calvo, R. (2008): A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. In: Computers & Operations Research 35, 1725-1741.

Elbek, M.; Wøhlk, S. (2016): A variable neighborhood search for the multi-period collection of recyclable materials. In: European Journal of Operational Research 249, 540-550.

Glover, F. (1986): Future paths for integer programming and links to artificial intelligence. In: Computers & Operations Research 13, 533-549.

Helsgaun, K. (2000): An effective implementation of the Lin-Kernighan traveling salesman heuristic. In: European Journal of Operational Research 126, 106-130.

Henke, T.; Speranza, M.G.; Wäscher, G. (2015): The multi-compartment vehicle routing problem with flexible compartment sizes. In: European Journal of Operational Research 246, 730-743.

Hübner, A.; Ostermeier, M. (2017): A multi-compartment vehicle routing problem with loading and unloading costs. In: Transportation Science, forthcoming.

Lahyani, R.; Coelho, L.C.; Khemakhem, M.; Laporte, G.; Semet, F. (2015): A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. In: Omega 51, 1-10.

Laporte, G. (2009): Fifty years of vehicle routing. In: Transportation Science 43, 408-416.

Lin, S.; Kernighan, B.W. (1973): An effective heuristic algorithm for the traveling-salesman problem. In: Operations Research 21, 498-516.

Popović, D.; Vidović, M.; Radivojević, G. (2012): Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. In: Expert Systems with Applications 39, 13390-13398.

Toth, P.; Vigo, D. (2014): Vehicle Routing: Problems, Methods, and Applications (2nd ed.). Society for Industrial and Applied Mathematics, Philadelphia.

Vidović, M.; Popović, D.; Ratković, B. (2014): Mixed integer and heuristics model for the inventory routing problem in fuel delivery. In: International Journal of Production Economics 147, 593-604.

VII

Summary and Outlook on Further Research Opportunities

Summary and Outlook on Further Research Opportunities

This thesis has focused on multi-compartment vehicle routing problems arising in the context of glass waste collection. In addition to a general classification of multi-compartment vehicle routing problems and a detailed overview on the current state of research, this thesis introduced solution approaches for several problem variants which have been considered for the first time. The regarded problem variants differ with respect to two characteristics: (1) The type of flexibility of the compartments, and (2) the length of the planning horizon. Vehicles with flexible compartments and problems with multiple periods have rarely been considered in the literature before. Hence, the analyses within this thesis provide valuable insights into the benefits of different types of compartment flexibility and the impact of varying lengths of the considered planning horizon.

For a problem with discretely-flexible compartment sizes and a single planning period, a heuristic solution approach, namely a variable neighborhood search, and an exact solution approach, namely a branch-and-cut approach, have been introduced. Moreover, for a slightly modified problem with continuously-flexible compartment sizes a heuristic, namely a genetic algorithm, has been developed. Finally, for a problem with multiple periods, a matheuristic approach has been proposed. All algorithms have been extensively tested by means of numerical experiments in order to analyze their performances with respect to solution quality and computing time. Additionally, several valuable managerial insights on multi-compartment vehicle routing problems have been obtained. With respect to the benefits of multi-compartment vehicles, a significant potential of reducing operational costs by 34.8% on average has been identified if vehicles with two compartments instead of vehicles with one compartment are used. Experiments with respect to different types of compartment flexibility have indicated only small reductions in operational costs when continuously-flexible compartment are used compared to discretely-flexible compartments. Finally, the consideration of multiple planning periods instead of a single planning period within the planning phase lead to cost reductions of 4.5%, on average, for the considered data sets. Clearly, these findings demonstrate the managerial benefits of using vehicles with multiple compartments and considering multiple panning periods, simultaneously, when vehicle routing problems with multiple product types are regarded.

Although many aspects of the real-world glass waste collection problem have been taken into account within the studies of this thesis, there are several aspects which have been neglected so far. Probably the most significant neglected aspect is the consideration of stochasticity with respect to the glass waste quantities. In the real-world problem, the amount of glass waste to be collected from a container is usually not known before the container is emptied. Therefore, determining vehicle tours when quantities are assumed to be deterministic might result in solutions which cannot be implemented. If, for example, the amount of glass waste in some containers was higher than estimated, the vehicle capacity might not be sufficient to empty all containers assigned to the respective tour. Instead, the vehicle would be forced to terminate its planned tour early in order to return to the depot and to empty its compartments. Only afterwards it would be able to continue its tour. To illustrate the consequences of considering deterministic quantities, a simplified example with one depot (D), one vehicle with a capacity of four capacity units, two container locations (C1 and C2), and two glass waste types is depicted in Figure 1. In this example, the estimated (deterministic) collection quantities amount to one capacity unit for each container, whereas for one container the observed collection quantity amounts to two capacity units (see green container at C2). A solution to this example for the deterministic case is depicted in Figure 1a). This solution could, however, not be implemented as the actual amount of glass waste at location C2 is higher than estimated. Consequently, after having identified the actual quantities at location C2, the collection vehicle would need to return to the depot, and after having emptied its compartments, the vehicle would perform an additional tour visiting location C2, only. This resulting situation is depicted in Figure 1b). In contrast, if the stochasticity of the collection quantities would already be considered during the planning phase, a more reliable solution might be obtained. Reliability in this context refers to a solution which might remain realizable even if the observed collection quantities differed slightly from the estimated quantities. A more reliable solution to the introduced example is shown in Figure 1c). In this solution, each location is visited on a single tour, thus, resulting in an estimated capacity utilization of 50% for each tour, instead of an estimated capacity utilization of 100% for the deterministic solution. For this example, it can be observed that solution c) would result in a smaller travelled distance than solution b), thus, justifying a consideration of stochasticity in the planning problem.

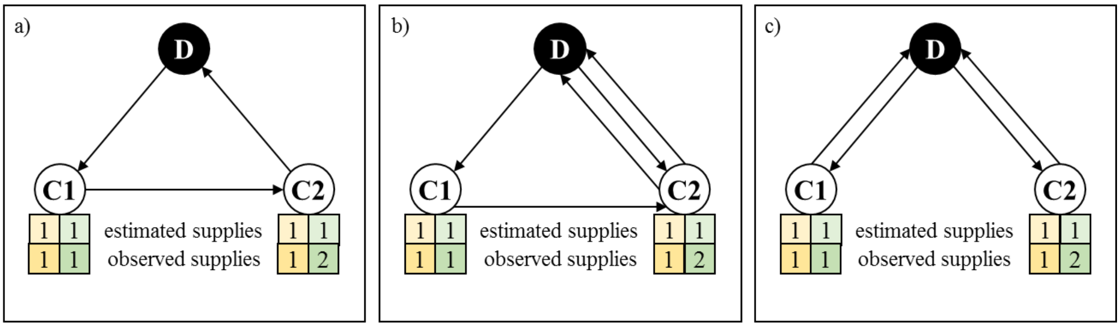


Figure 1: Example for a multi-compartment vehicle routing problem with stochastic supplies

Stochastic delivery or collection quantities in vehicle routing problems have already been extensively studied in the literature (see e.g. Gendreau et al., 2014). Moreover, some first examinations on multi-compartment vehicle routing problems with stochastic demands have also been presented (see e.g. Mendoza et al., 2010), all of which consider fixed compartments and assignments of product types to compartments. Both of these characteristics are, however, opposite in the glass waste collection problem. Therefore, an investigation of the impact of stochasticity on multi-compartment vehicle routing problems with flexible compartments and flexible assignments provides a promising opportunity for future research. A comparison between deterministic and stochastic collection quantities might further generate managerial findings on the value of perfect information in this context. Such insights might be especially interesting as recently some new technologies have been developed which allow for monitoring the fill rates of waste containers (Gutierrez et al., 2015). Thus, knowing about the value of perfect information would provide companies with a good basis for deciding about acquiring such technologies.

Another option to increase the efficiency of collection tours represents the consideration of additional capacities within the vehicles, e.g. by using trailers which can be coupled to the main collection vehicle. By doing so, not only the collection capacity, but also the number of compartments for a collection vehicle could be increased. However, adding a trailer to a vehicle has the disadvantage of increasing the vehicle's overall size and weight. Thus, limitations with respect to the accessibility of roads and container locations need to be taken into account. Vehicle routing problems which consider a truck and trailer component have already been discussed in the literature in other contexts (Li et al., 2016). A feasible vehicle route in such planning problems often consists of a main route performed by the truck with trailer which serves easily accessible customers, and some diverging sub-routes performed by the truck, only, which serve customers with limited access. A multi-compartment vehicle routing problem with a truck and trailer component has already been presented by Caramia and Guerriero (2010) for fixed compartments and fixed assignments of product types to compartments. A study considering the specifications of the glass waste collection problem would provide further insights into the benefits and limitations of increasing vehicles' capacities by means of trailers.

A further component from the real-world glass waste collection problem, which has so far been neglected, is the existence of different container models with respect to their mechanical requirements. In the real-world problem, some containers are placed above ground, whereas other containers, usually located in densely populated areas, are installed below ground. As a consequence, these different container models require different mechanical equipment to be lifted by the collection vehicles. Although, the equipment can be changed from container site to container site, a certain handling time is required for such an operation. So far, this aspect has seemingly not been considered in the literature. Therefore, a detailed analysis of this problem extension provides another interesting opportunity for future research.

Finally, also the location of the containers themselves as well as their capacities could be subject to further adjustments. Often, the decision about container locations and capacities is made without considering the consequences on the vehicle routing problems. However, integrating both decision components into a single planning problem might lead to overall more efficient collection tours. Problems which combine routing and location decisions are known as location routing problems and have already been extensively studied in the literature (see e.g. Drexl and Schneider, 2015). The work by Hemmelmayr et al. (2014) even considers such a problem in a waste collection context. However, they do not take vehicles with multiple compartments for the collection operations into account. Thus, the authors' findings cannot necessarily be generalized to the glass waste collection problem. Therefore, an investigation of container locations and container sizes for problems with multiple compartments provides another option for further research efforts.

Concluding, the problem variants considered in this thesis already cover many aspects of the real-world glass waste collection problem. Still, there are several further problem modifications and extensions which have not been studied, so far, and which offer interesting research questions for future projects.

References

Caramia, M.; Guerriero, F. (2010): A milk collection problem with incompatibility constraints. In: Interfaces 40, 130-143.

Drexl, M.; Schneider, M. (2015): A survey of variants and extensions of the location-routing problem. In: European Journal of Operational Research 241, 283-308.

Gendreau, M.; Jabali, O.; Rei, W. (2014): Stochastic vehicle routing problems. In: Toth, P; Vigo, D. (eds.): Vehicle routing: problems, methods, and applications (2nd ed.). Philadelphia: Society for Industrial and Applied Mathematics, 213-239.

Gutierrez, J.M.; Jensen, M.; Henius, M.; Riaz, T. (2015): Smart waste collection system based on location intelligence. In: Procedia Computer Science 61, 120-127.

Hemmelmayr, V.C.; Doerner, K.F.; Hartl, R.F.; Vigo, D. (2014): Models and algorithms for the integrated planning of bin allocation and vehicle routing in solid waste management. In: Transportation Science 48, 103-120.

Li, H.; Lv, T.; Lu, Y. (2016): The combination truck routing problem: a survey. In: Procedia Engineering 137, 639-648.

Mendoza, J.E.; Castanier, B.; Guéret, C.; Medaglia, A.L.; Velasco, N. (2010): A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. In: Computers & Operations Research 37, 1886-1898.