



KATHOLISCHE UNIVERSITÄT
EICHSTÄTT-INGOLSTADT

**Modelle und Metaheuristiken für neuartige
Tourenplanungsprobleme**

Models and Metaheuristics for Emerging Vehicle Routing Problems

Markus Frank

Kumulative Dissertation
zur Erlangung des akademischen Grades
Doctor rerum politicarum (Dr. rer. pol.)

Tag der mündlichen Prüfung: 02. Juni 2022

Referent: Prof. Dr. Heinrich Kuhn
Korreferent: Prof. Dr. Michael Sternbeck

Zusammenfassung

Die vorliegende kumulative Dissertation behandelt Fragestellungen zu verschiedenen, innovativen Varianten des klassischen Tourenplanungsproblems. Sie umfasst die folgenden vier Beiträge:

1. Frank, M., Ostermeier, M., Holzapfel, A., Hübner, A., Kuhn, H., 2021. **Optimizing Routing and Delivery Patterns with Multi-compartment Vehicles**. European Journal of Operational Research, Volume 293, Issue 2, Pages 495-510 (accepted).
2. Voigt, S., Frank, M., Fontaine, P., Kuhn, H., 2021. **The Vehicle Routing Problem with Availability Profiles**. Transportation Science (Minor Revision).
3. Voigt, S., Frank, M., Fontaine, P., Kuhn, H., 2022. **Hybrid Adaptive Large Neighborhood Search for Vehicle Routing Problems with Depot Location Decisions**. Computers and Operations Research, Volume 146, 105856 (accepted).
4. Frank, M., 2022. **A Review on Emerging Variants of the Multi-period Vehicle Routing Problem**. Available at SSRN.

Die veröffentlichten bzw. eingereichten Versionen dieser Beiträge können aus Gründen der Konsistenz (z.B. Rechtschreibung, Nomenklatur) geringfügig von den Versionen in dieser Arbeit abweichen. Dies hat keinen Einfluss auf den Inhalt der angenommenen Beiträge. Der Inhalt von Arbeitspapieren hingegen kann sich während des Begutachtungsprozesses noch ändern.

Im ersten Beitrag wird zunächst ein taktisches Tourenplanungsproblem bei der regelmäßigen Belieferung von Filialen des Lebensmitteleinzelhandels über mehrere Planungsperioden hinweg untersucht. Anders als zuvor werden dabei Mehrkammerfahrzeuge verwendet, die eine gemeinsame Belieferung unterschiedlicher Produktgruppen ermöglichen. Dadurch können Synergieeffekte erzielt werden, wenn dies in die Planung der spezifischen Belieferungsmuster einbezogen wird. Als Lösungsmethode für dieses Planungsproblem wird eine Adaptive Large Neighborhood Search (ALNS) konzipiert. Im zweiten Beitrag wird das Tourenplanungsproblem um Zeitfenster und kundenindividuelle Anwesenheitswahrscheinlichkeiten erweitert, um die operative Auslieferung von Paketen auf der letzten Meile zu verbessern. Mit dem datengetriebenen Ansatz, der in die Planung der Auslieferungstouren einbezieht, wann die Empfänger eher zu Hause sind, kann die Zahl fehlschlagender Zustellungsversuche signifikant verringert werden. Um das hierzu entwickelte Modell effizient lösen zu können, wird außerdem die neuartige hybride ALNS (HALNS) entwickelt, die Elemente einer ALNS und genetischer Algorithmen vereint. Die HALNS wird im dritten Beitrag weiterentwickelt und auf eine ganze Klasse von Tourenplanungsproblemen angewandt, die zusätzlich zum klassischen Zuordnungs- und Reihenfolgeproblem auch Entscheidungen hinsichtlich der Standorte der verwendeten Depots treffen. Der vierte Beitrag schließlich spannt einen Bogen zurück zum mehrperiodischen Tourenplanungsproblem. Die bestehende Literatur zu seinen taktischen und operativen Varianten wird begutachtet und charakterisiert, wobei erstmalig beide Formen klar voneinander abgegrenzt werden. Die genannten Beiträge werden im Folgenden kurz zusammengefasst.

Beitrag 1 - Optimizing Routing and Delivery Patterns with Multi-compartment Vehicles

Einzelhändler nutzen typischerweise sich wöchentlich wiederholende Belieferungsmuster, u.a. für eine vereinfachte Personalplanung zur Regalbefüllung in der Filiale, für regelmäßige Transporttouren und für das Kapazitätsmanagement im Verteilzentrum. Dieses Vorgehen ermöglicht eine gemeinsame, einheitliche Planung der logistischen Subsysteme Filiale, Transport und Verteilzentrum innerhalb der internen Supply Chain des Einzelhändlers. Für Lebensmittel gilt es, während dieser Aktivitäten unterschiedliche Temperaturanforderungen zu beachten. Solange der Transport von Lebensmitteln mit verschiedenen Temperaturzonen deshalb mit unterschiedlichen (Kühl-)Fahrzeugen durchgeführt wurde, konnten die Belieferungen verschiedener Temperaturzonen nicht miteinander koordiniert werden. Die vor einiger Zeit eingeführten Mehrkammerfahrzeugen machen dies nun möglich, erfordern aber eine neue Planung der wöchentlichen Belieferungsmuster. Beispielsweise können nun Langsamdreher aus einer Temperaturzone gemeinsam mit Schnelldrehern einer anderen Temperaturzone transportiert werden und damit ebenfalls häufiger angeliefert werden.

Ziel ist es damit, für jedes temperaturspezifische Produktsegment einer jeden Filiale das optimale Belieferungsmuster (Anzahl der Lieferungen in einer Woche und die spezifischen Tage) auszuwählen, sowie die daraus entstehenden Auslieferungstouren festzulegen. Im Beitrag werden zunächst alle mit der Filialbelieferung zusammenhängenden Prozesse entlang der internen Supply Chain von Lebensmittelhändlern analysiert, um so alle zu beachtenden Restriktionen und die entscheidungsrelevanten Kosten definieren zu können. Diese können wiederum zu Kostenblöcken aggregiert werden. Bei einer gegebenen wöchentlichen Filialnachfrage besteht der Trade-Off hier zwischen häufigeren, kleineren Lieferungen, die Wiederverräumkosten und Lagerhaltung in der Filiale verringern, und selteneren, größeren Lieferungen, die Vorteile beim Kommissionieren im Verteilzentrum und beim Handling im Transport bieten. Zusätzlich können Transportkosten gespart werden, wenn die Belieferungstage zwischen Produktsegmenten und Filialen koordiniert werden. Diese Problemstellung wird als periodisches Vehicle Routing Problem mit Mehrkammerfahrzeugen (PMCVRP) formalisiert. Zum Erzeugen von Lösungen für das \mathcal{NP} -schwere PMCVRP wird eine Metaheuristik entwickelt, die in jeder Verfahrensiteration zwei Teilschritte durchführt. Die Zuweisung von Belieferungsmustern zu Filial-Segment Kombinationen geschieht durch eine ALNS, die praxismotivierte Operatoren verwendet, während die Lösung der dadurch entstehenden Tourenplanungsprobleme für jeden Tag mit Hilfe einer schnelleren LNS erfolgt.

Der Lösungsansatz wird zunächst zur Evaluation mit einem bestehenden Verfahren für ein ähnliches Problem verglichen. Die ALNS erzielt dabei signifikant bessere Ergebnisse. Im Folgenden wird der Ansatz auf synthetisch generierte Instanzen angewandt. Es kann gezeigt werden, dass Mehrkammerfahrzeuge gegenüber regulären Fahrzeugen Kostenvorteile bieten, die je nach Nachfrageverteilung der Produktsegmente bis zu 15% betragen. Zum Abschluss zeigt eine große Fallstudie mit empirischen Daten eines Lebensmittelhändlers, dass mit Hilfe des verwendeten Planungsansatzes etwa 8% Kosteneinsparungen erzielt werden könnten sowie die durchschnittliche Fahrzeugauslastung um ca. 8% erhöht werden könnte.

Beitrag 2 - The Vehicle Routing Problem with Availability Profiles

Der zweite Beitrag behandelt ein Tourenplanungsproblem bei der Haustürzustellung von Paketen. In vielen Fällen ist die Anwesenheit des empfangenden Kunden bei der Zustellung unbedingt erforderlich. Ist er zu diesem Zeitpunkt nicht zuhause, schlägt die Zustellung fehl, was zusätzliche Kosten für den Paketdienstleister und Unzufriedenheit beim Kunden verursacht. Paketdienstleister berichten von verhältnismäßig hohen Raten fehlschlagender Zustellungen, unter anderem aufgrund fehlender Zeitfenster für die Zustellung. Das Einführen solcher fixen Zeitfenster führt aber einerseits zu weniger effizienten Zustelltouren und ist andererseits aufgrund fehlender Kontaktmöglichkeiten häufig nicht möglich. Als Alternative wird vorgeschlagen, kundenbezogene Daten, bspw. aus der Historie vergangener Zustellungen, zu nutzen, um Wahrscheinlichkeiten für die Anwesenheit des Kunden abzuleiten. Diese werden zu kundenindividuellen Verfügbarkeitsprofilen aggregiert, die einen kompletten Belieferungstag umfassen und aus Zeitfenstern mit den jeweiligen Anwesenheitswahrscheinlichkeiten bestehen. Die Kosten für eine fehlgeschlagene Belieferung hängen zudem von der Politik des Paketdienstleisters ab, z.B. ob eine Zustellung zum Nachbar durchgeführt wird. Gemeinsam mit den Verfügbarkeitsprofilen können so die erwarteten Kosten für fehlgeschlagene Belieferungen in Abhängigkeit der gewählten Zeitfenster berechnet werden.

Das Entscheidungsproblem, das optimale Zeitfenster bzw. den optimalen Zustellzeitpunkt für jeden Kunden auszuwählen und dabei den Trade-Off zwischen den erwarteten Kosten für fehlgeschlagene Zustellungen und den Transportkosten zu beachten, wird als Vehicle Routing Problem mit Verfügbarkeitsprofilen (VRPAP) modelliert. Das VRPAP ist eine Generalisierung des bekannten Vehicle Routing Problem mit einem oder mehreren Zeitfenstern (VRPTW, VRPMTW) und als solches \mathcal{NP} -schwer. Um effizient Lösungen zu generieren wird deshalb eine neuartige Metaheuristik entwickelt, eine hybride Adaptive Large Neighborhood Search (HALNS), die eine adaptive große Nachbarschaftssuche (ALNS) mit Elementen genetischer Algorithmen kombiniert. Die ALNS wird dabei verwendet, eine Population diverser Lösungen zu erzeugen, die in den folgenden Verfahrensiterationen parallel durch Entfernen- und Einfügeoperatoren weiterverbessert werden sollen. Dabei wird die aktuell beste Lösung der Population zum Vorbild genommen, indem Kunden, die dort gleich platziert sind nur mit geringer Wahrscheinlichkeit entfernt werden.

Bei der Anwendung auf das VRPTW mit harten und weichen Zeitfenstern und das VRPMTW liefert das Lösungsverfahren Ergebnisse, die mit den aktuell besten Algorithmen vergleichbar sind. Insbesondere für die letzteren beiden VRP-Varianten können zahlreiche neue beste bekannte Lösungen gefunden werden. Basierend auf theoretischen Verfügbarkeitsprofilen werden auch neue VRPAP-Instanzen erzeugt, die in weiterführenden Experimenten verwendet werden. Diese zeigen, dass insbesondere die Zahl der Peaks und ein geeigneter Mix von Profilen zu Kostenersparnissen beitragen. Die Zustellrate kann durch den VRPAP-Ansatz bereits zu geringen zusätzlichen Transportkosten erheblich gesteigert werden. Schließlich wird mithilfe von empirischen Daten zur Kundenanwesenheit eine realitätsnahe Fallstudie konzipiert und untersucht. Hier können alleine die beiden Kundenmerkmale Alter und Erwerbstätigkeit ausreichen um Verfügbarkeitsprofile zu erzeugen, deren Einbeziehung in die Tourenplanung die Zustellrate um 12% und die Gesamtkosten um 5% verbessert.

Beitrag 3 - Hybrid Adaptive Large Neighborhood Search for Vehicle Routing Problems with Depot Location Decisions

Im dritten Beitrag wird die zuvor vorgestellte HALNS auf Varianten des Tourenplanungsproblems angewandt und adaptiert, bei denen zusätzlich zum klassischen Reihenfolge- und Zuordnungsproblem Kunden auch einem von mehreren möglichen Depots zugeordnet werden müssen. Diese Problemklasse umfasst das Two-Echelon Vehicle Routing (2E-VRP) Problem, das Location Routing Problem (LRP) und das Multi-Depot Vehicle Routing Problem (MDVRP). Es wird gezeigt, dass LRP und MDVRP als Spezialfälle des 2E-VRP modelliert werden können, so dass eine Lösung mit einem einheitlichen Verfahren sinnvoll sein kann. Mit einer fortentwickelten HALNS wird ein solches Verfahren konzipiert. Wie im zweiten Beitrag wird eine Population von Lösungen erzeugt und im Laufe des Verfahrens parallel weiterentwickelt, so dass mit hoher Wahrscheinlichkeit Lösungen mit vielen unterschiedlichen Standortkonfigurationen untersucht werden. Die einzelnen Lösungen werden über eine möglichst effiziente ALNS generiert und verbessert, wobei sich die Operatoren wiederum an der aktuell besten Lösung der Population orientieren, anstatt einen klassischen Kreuzungsmechanismus zu verwenden. Ähnlich wie in klassischen genetischen Algorithmen überleben Lösungen allerdings nicht nur aufgrund ihrer Güte, sondern auch aufgrund ihres Beitrags zur Diversifizierung der Population. Da mit dem Problem der Depotstandorte eine zusätzliche Entscheidung einer hierarchisch übergeordneten Stufe hinzukommt, wird das Verfahren um eine zusätzliche Heuristik für speziell diese Stufe sowie um ein Tabuverfahren für Depotstandorte erweitert.

Einen Beitrag zur bestehenden Literatur liefert diese Arbeit aber auch aufgrund eines umfassenden Überblicks zu bestehenden Lösungsverfahren für die drei Problemstellungen und deren Leistungsfähigkeit. Die vorgestellten Experimente mit den jeweils üblichen Benchmarkinstanzen aus der Literatur zeigen für die HALNS ähnlich gute Ergebnisse wie spezialisierte Heuristiken und damit, dass die HALNS das Ziel eines einheitlichen Lösungsverfahrens erreicht hat. Anders als in der Literatur häufig vertreten, zeigt dies, dass breiter angelegte Verfahren nicht unbedingt schlechtere Lösungen liefern. Durch die Hybridisierung kann die Varianz über mehrere Verfahrensläufe hinweg außerdem vergleichbar niedrig gehalten werden. Im Übrigen werden drei neue beste bekannte Lösungen für das LRP gefunden.

Beitrag 4 - A Review on Emerging Variants of the Multi-period Vehicle Routing Problem

Eine der am häufigsten vorgenommenen Erweiterungen des klassischen Tourenplanungsproblems (VRP) ist seine Generalisierung auf mehrere Zeitperioden, denn viele Entscheidungsprobleme, die als VRP-Variante modelliert werden, beinhalten Entscheidungen, die mehrere Perioden umfassen oder sich auf diese aufteilen. Innerhalb des Forschungsgebiets der mehrperiodischen VRPs lag der Schwerpunkt in der Vergangenheit vor allem auf Anwendungen, bei denen Kunden über mehrere Perioden zyklisch wiederkehrend besucht werden müssen. Das sog. periodische VRP (PVRP) kann aber als Spezialfall des generischeren mehrperiodischen VRPs (MPVRP) gesehen werden, das in der Literatur ebenfalls Erwähnung findet. Im Vergleich zum PVRP ist die Forschung zum nicht-periodischen MPVRP weniger stark ausgeprägt, obwohl derartige Problemstellungen ebenfalls häufig auftreten, insbesondere in der dynamischen Modellierung und bei praxisnahen Anwendungen, wo

eine Vielzahl von Nebenbedingungen beachtet werden muss. Bisher wurden die beiden Begrifflichkeiten MPVRP und PVRP in der Literatur allerdings nicht trennscharf voneinander abgegrenzt, beide Terme werden mit unterschiedlichem Verständnis und einander ersetzend verwendet. Der vierte Beitrag setzt an diesem Punkt an, greift die unterschiedlichen Verwendungsweisen auf und nutzt schließlich das Kriterium der Periodizität, um Modellformulierungen klar dem PVRP oder MPVRP zuzuordnen. Weiterhin präsentiert er eine erste Modellformulierung für ein allgemeines MPVRP und seine häufigsten Erweiterungen, wie die Einführung von periodenbezogenen Servicekosten. Für den folgenden Literaturüberblick wird zunächst eine Typologisierung von VRPs mit mehreren Perioden entwickelt, die sich explizit an den Charakteristika mehrperiodischer VRPs orientiert. Diese lassen sich unter den Oberpunkten Periodizität, Besuchsfrequenz, Einschränkungen der Besuchstage und zielfunktionsbezogene Komponenten subsumieren. Im Literaturüberblick wird die Klasse der MPVRPs zudem in statische und dynamische Problemstellungen gegliedert. Insgesamt werden 54 Publikationen charakterisiert und diskutiert, sowie einige zusammenhängende Literaturströme aufgezeigt.

Zusammenfassung

Die vorliegende kumulative Dissertation beschäftigt sich sowohl problemorientiert als auch methodisch mit Tourenplanungsproblemen. Aus problemorientierter Sicht werden einerseits neu auftretende Planungsprobleme aus der Praxis untersucht, abstrahiert und modelliert, zum einen für die taktische Filialbelieferung mit Mehrkammerfahrzeugen und zum anderen für die operative Auslieferung von Paketen unter Berücksichtigung von Verfügbarkeitsprofilen (Beitrag 1 und 2). Andererseits werden verwandte Problemstellungen aus der bestehenden Literatur typologisiert und diskutiert (Beitrag 4). Auf methodischer Ebene werden metaheuristische Lösungsverfahren entwickelt bzw. angepasst, um diese auf die zuvor konzipierten oder auf bekannte Tourenplanungsprobleme anzuwenden (Beitrag 1, 2 und 3).

Abstract

This cumulative dissertation addresses issues related to different, innovative variants of the classical vehicle routing problem (VRP). It comprises the following four contributions:

1. Frank, M., Ostermeier, M., Holzapfel, A., Hübner, A., Kuhn, H., 2021. **Optimizing Routing and Delivery Patterns with Multi-compartment Vehicles**. European Journal of Operational Research, Volume 293, Issue 2, Pages 495-510 (accepted).
2. Voigt, S., Frank, M., Fontaine, P., Kuhn, H., 2021. **The Vehicle Routing Problem with Availability Profiles**. Transportation Science (Minor Revision).
3. Voigt, S., Frank, M., Fontaine, P., Kuhn, H., 2022. **Hybrid Adaptive Large Neighborhood Search for Vehicle Routing Problems with Depot Location Decisions**. Computers and Operations Research, Volume 146, 105856 (accepted).
4. Frank, M., 2022. **A Review on Emerging Variants of the Multi-period Vehicle Routing Problem**. Available at SSRN.

The published or submitted versions of these manuscripts may differ slightly from the versions in this thesis for consistency (e.g., spelling, nomenclature). This does not affect the content of accepted manuscripts. In contrast, the content of manuscripts submitted but not yet accepted is subject to changes during the review process.

In the first paper, we investigate a tactical VRP for regular deliveries to grocery retail stores over several planning periods. In contrast to previous approaches, multi-compartment vehicles are used, which allow joint deliveries of different product groups. Thus, synergy effects can be achieved if this is included in the planning of the specific delivery patterns. As a heuristic solution method for this planning problem, an Adaptive Large Neighborhood Search (ALNS) is designed. In the second paper, the VRP is extended to include time windows and customer-specific availability probabilities to improve the operational delivery of last-mile parcels. The data-driven approach that incorporates information on when recipients are more likely to be at home into delivery route planning can significantly reduce the number of failed delivery attempts. In order to efficiently solve the model developed for this purpose, the novel hybrid ALNS (HALNS), which combines elements of an ALNS and genetic algorithms. The HALNS is further developed in the third paper and applied to a whole class of VRPs that, in addition to the classical assignment and sequencing problem, also make decisions regarding the locations of the depots used. Finally, the fourth paper focuses on the multi-period VRP. The existing literature on its tactical and operational variants is surveyed and characterized, clearly distinguishing the two forms for the first time.

Contents

| | |
|--|-------------|
| List of Figures | xii |
| List of Tables | xiii |
| 1 Optimizing Routing and Delivery Patterns with MCVs | 2 |
| 1.1 Introduction & Motivation | 3 |
| 1.2 Grocery Supply Chain, associated Processes and Costs | 4 |
| 1.2.1 Distribution Processes in Grocery Retailing | 4 |
| 1.2.2 Selection of Delivery Patterns | 5 |
| 1.2.3 Identification of decision-relevant Costs and Constraints for defining DPs and using MCVs | 6 |
| 1.3 Related Literature | 9 |
| 1.4 Decision Model for Periodic Multi-compartment Vehicle Routing Problem | 11 |
| 1.5 Solution Approach | 14 |
| 1.5.1 Initial Solution | 15 |
| 1.5.2 Improvement Heuristic | 15 |
| 1.5.2.1 Stage 1: ALNS for Optimizing DPs | 15 |
| 1.5.2.2 Stage 2: LNS for Solving the Routing in each Period | 20 |
| 1.5.3 Evaluation and Control Mechanism of entire Algorithm | 21 |
| 1.6 Numerical Experiments | 22 |
| 1.6.1 Runtime Analysis | 24 |
| 1.6.2 Comparison with Single Segment Benchmark | 25 |
| 1.6.3 Planning of DPs across Segments with MCVs | 27 |
| 1.6.4 Case Study | 29 |
| 1.7 Conclusion | 31 |
| 2 VRP with Availability Profiles | 34 |
| 2.1 Introduction & Motivation | 35 |
| 2.2 Problem Description | 36 |
| 2.2.1 Delivery Process | 36 |
| 2.2.2 Failed Deliveries & Availability Profiles | 36 |
| 2.2.3 Decision-relevant Costs and Constraints | 37 |
| 2.2.4 Operational Planning Problem | 38 |
| 2.3 Related Literature | 38 |
| 2.3.1 Time-constrained Vehicle Routing Literature | 39 |
| 2.3.1.1 VRPTW | 39 |

Contents

| | | |
|----------|--|-----------|
| 2.3.1.2 | VRPMTW | 40 |
| 2.3.1.3 | VRPSTW | 40 |
| 2.3.1.4 | VRPGTW | 41 |
| 2.3.2 | Failed Delivery Literature | 41 |
| 2.3.3 | Summary and Research Gap | 42 |
| 2.4 | Decision Model for Vehicle Routing with Availability Profiles | 43 |
| 2.5 | Hybrid Adaptive Large Neighborhood Search | 45 |
| 2.5.1 | Solution Representation and Penalized Costs | 46 |
| 2.5.2 | ALNS Algorithm | 47 |
| 2.5.2.1 | Information Collection | 48 |
| 2.5.2.2 | Determination of Removal Candidates | 48 |
| 2.5.2.3 | Removal Operators | 48 |
| 2.5.2.4 | Determination of Insertion Order | 50 |
| 2.5.2.5 | Insertion Operators | 50 |
| 2.5.2.6 | Simulated Annealing | 50 |
| 2.5.2.7 | Adaptivity | 51 |
| 2.5.3 | Crossover Phase | 51 |
| 2.5.4 | Selection of Survivors | 51 |
| 2.6 | Numerical Experiments | 51 |
| 2.6.1 | VRPAP Instance Generation | 52 |
| 2.6.2 | Performance Evaluation | 52 |
| 2.6.2.1 | Benchmark for Time-Constrained VRPs | 53 |
| 2.6.2.2 | Benchmark for VRPAP Instances | 54 |
| 2.6.2.3 | Analysis of Algorithmic Components | 56 |
| 2.6.3 | Sensitivity Analysis and Managerial Insights | 57 |
| 2.6.3.1 | Cost Savings Potential | 57 |
| 2.6.3.2 | Analysis of Availability Profiles | 59 |
| 2.6.3.3 | Analysis of Solution Structure | 59 |
| 2.6.3.4 | Distance Costs vs. Failed-Delivery Rate | 60 |
| 2.6.3.5 | Effect of Policies for Failed Deliveries | 61 |
| 2.6.4 | Case Study with Empirical Availability Profiles | 62 |
| 2.7 | Conclusions and Future Areas of Research | 63 |
| 3 | HALNS for VRPs with Depot Location Decisions | 66 |
| 3.1 | Introduction | 67 |
| 3.2 | Vehicle Routing with Depot Location Decisions | 68 |
| 3.2.1 | 2-Echelon VRP | 68 |
| 3.2.2 | Location Routing Problem and Multi-Depot VRP as Special Cases of the 2E-VRP | 69 |
| 3.3 | Overview of Related Heuristics | 71 |
| 3.3.1 | 2-Echelon VRP | 71 |

| | | |
|----------|--|-----------|
| 3.3.2 | Location Routing Problem | 72 |
| 3.3.3 | Multi-Depot VRPs | 73 |
| 3.3.4 | Summary | 74 |
| 3.4 | Hybrid Adaptive Large Neighborhood Search for VRPs with Depot Location Decisions | 77 |
| 3.4.1 | ALNS Algorithm | 77 |
| 3.4.1.1 | Information Collection | 79 |
| 3.4.1.2 | Determination of Removal Candidates | 79 |
| 3.4.1.3 | Removal Operators | 79 |
| 3.4.1.4 | Determination of Insertion Order | 80 |
| 3.4.1.5 | Insertion Operators | 80 |
| 3.4.1.6 | First-Level Heuristic | 81 |
| 3.4.1.7 | Local Improvement of Routes | 82 |
| 3.4.1.8 | Accounting for Symmetries | 82 |
| 3.4.1.9 | Simulated Annealing | 83 |
| 3.4.1.10 | Adaptive Parameters | 83 |
| 3.4.2 | Crossover and Education Phase | 83 |
| 3.4.3 | Selection of Survivors and Diversity Management | 84 |
| 3.4.4 | Setting Depot Locations Tabu | 84 |
| 3.5 | Numerical Experiments | 85 |
| 3.5.1 | Instances and Experimental Setting | 85 |
| 3.5.2 | Parameter Tuning | 86 |
| 3.5.3 | Benchmarks | 86 |
| 3.5.3.1 | 2E-VRP Benchmark | 87 |
| 3.5.3.2 | LRP Benchmark | 88 |
| 3.5.3.3 | MDVRP Benchmark | 89 |
| 3.5.4 | Analysis of Algorithm Components | 90 |
| 3.6 | Summary and Further Areas of Research | 94 |
| 4 | A review on the multi-period VRP | 97 |
| 4.1 | Introduction | 98 |
| 4.2 | A Definition of the Multi-period Vehicle Routing Problem | 99 |
| 4.2.1 | “PVRP” versus “MPVRP” | 99 |
| 4.2.2 | A General Multi-period Vehicle Routing Problem and its Formulation | 100 |
| 4.3 | Literature Review | 102 |
| 4.3.1 | Literature on the MPVRP | 103 |
| 4.3.1.1 | A Typology for Multi-period Vehicle Routing Problems | 103 |
| 4.3.1.2 | Static MPVRP | 104 |
| 4.3.1.3 | Dynamic MPVRP | 106 |
| 4.3.2 | Literature on the PVRP | 108 |
| 4.3.3 | Solution Approaches for VRPs with Multiple Periods | 111 |
| 4.4 | Conclusion | 111 |

| | |
|---|------------|
| Bibliography | 113 |
| A Appendix for VRPAP | 124 |
| A.1 Parameters | 124 |
| A.2 Results for VRPTW, VRPMTW and VRPSTW Benchmark Instances | 125 |
| A.2.1 Results for Solomon and Desrosiers (1988) VRPTW Instances | 125 |
| A.2.2 Results for Belhaiza et al. (2014) VRPMTW Instances | 127 |
| A.2.3 Results for VRPSTW Instances | 130 |
| A.2.4 Results for VRPAP Instances | 132 |
| A.3 Results for Components | 133 |
| A.4 UKTUS Availability Profiles | 133 |
| B Appendix for HALNS for VRPs with Depot Location Decisions | 135 |
| B.1 Parameters | 135 |
| B.2 Scaling of Runtimes | 135 |
| B.3 Detailed Results on Benchmark Instances | 136 |
| Ehrenwörtliche Erklärung | xiv |

List of Figures

| | | |
|------|---|-----|
| 1.1 | Distribution process with MCVs | 5 |
| 1.2 | Example of delivery patterns for a store with three segments | 6 |
| 1.3 | Decision-relevant operations, distribution costs and constraints along the internal retail supply chain | 8 |
| 1.4 | Algorithmic structure | 14 |
| 1.5 | Distribution of total cost savings depending on delivery area and number of stores in comparison to the adapted benchmark of Holzapfel et al. (2016) (best of 5 runs) | 26 |
| 2.1 | Alternative constellations of customer availability across the delivery horizon | 39 |
| 2.2 | Total costs depending on c^{failed} and the number of vehicles used | 58 |
| 2.3 | Cost savings and cost distributions in delivery planning depending on APs | 59 |
| 2.4 | Solution structure depending on failed-delivery cost parameter c^{failed} for C101-WM | 60 |
| 2.5 | Failed-delivery rate vs. increase distance cost | 61 |
| 3.1 | Exemplary 2E-VRP instance | 69 |
| 3.2 | Exemplary LRP instance as special case of the 2E-VRP | 70 |
| 3.3 | Exemplary MDVRP instance as special case of the 2E-VRP | 70 |
| 3.4 | Schematic overview of a single HALNS iteration | 78 |
| 3.5 | Search trajectories for different α on subset of 2E-VRP instances | 87 |
| 3.6 | Performance chart 2E-VRP | 88 |
| 3.7 | Performance chart LRP | 89 |
| 3.8 | Performance chart MDVRP | 90 |
| 3.9 | Analysis of components of HALNS for 2E-VRP | 93 |
| 3.10 | 2E-VRP low-impact components are high-impact components for MDVRP | 94 |
| 4.1 | Attributes of MPVRPs. | 104 |
| 4.2 | Classification of allowable visit periods. | 109 |
| A.1 | VRPTW Benchmark | 125 |
| A.2 | VRPMTW Benchmark | 127 |
| A.3 | HALNS performance with and without specific components for VRPAP instances | 133 |
| A.4 | UKTUS Availability Profiles depending on Age and Work Status | 134 |

List of Tables

| | | |
|------|---|----|
| 1.1 | Notation used to model PMCVRP | 11 |
| 1.2 | Operators used within the ALNS algorithm | 16 |
| 1.3 | Overview of numerical experiments | 23 |
| 1.4 | Algorithm parameters used | 24 |
| 1.5 | Setting of parameter c | 24 |
| 1.6 | Total computation times for different problem sizes, in hours | 25 |
| 1.7 | Cost savings in comparison to the adapted benchmark of Holzapfel et al. (2016) (best of 5 runs) | 26 |
| 1.8 | (Mean) segment share of total order volume | 28 |
| 1.9 | Share of costs components of total costs | 28 |
| 1.10 | Cost savings of MCV compared to SCV | 28 |
| 1.11 | Solution structure of MCV compared to SCV, entire planning period | 29 |
| 1.12 | Cost savings of the integrated approach compared to status quo | 30 |
| 1.13 | Solution structure resulting from the integrated approach compared to the status quo approach | 30 |
| 2.1 | Notation | 43 |
| 2.2 | Experiments and data sets | 52 |
| 2.3 | Availability profiles | 52 |
| 2.4 | Summarized results for Solomon (1987) VRPTW instances | 53 |
| 2.5 | Summarized results for Belhaiza et al. (2014) VRPMTW instances | 54 |
| 2.6 | Summarized results for VRPSTW instances type 1 (only lateness), $\alpha = 1$ | 55 |
| 2.7 | Summarized results for VRPSTW instances type 2 (earliness and lateness), $\alpha = 1$ | 55 |
| 2.8 | Summarized results for VRPAP instances | 56 |
| 2.9 | Influence of specific components on HALNS performance | 57 |
| 2.10 | VRPAP vs. VRPMTW | 58 |
| 2.11 | Results on VRPAP instances with neighbor delivery for failed deliveries | 61 |
| 2.12 | Average Results on Solomon Instances with UK TUS APs | 63 |
| 3.1 | Classification of heuristics solving 2E-VRP, LRP, or MDVRP | 76 |
| 3.2 | Instances | 85 |
| 3.3 | Parameter tuning for HALNS on 2E-VRP instances | 86 |
| 3.4 | Heuristics for 2E-VRP | 88 |
| 3.5 | Heuristics for LRP | 89 |
| 3.6 | Heuristics for MDVRP | 90 |
| 3.7 | Analysis of HALNS solutions of 2E-VRP when disabling one component at a time | 91 |

List of Tables

| | | |
|-----|---|-----|
| 4.1 | Legend for Tables 4.2-4.3 | 104 |
| 4.2 | Characterization of Static MPVRPs | 105 |
| 4.3 | Characterization of Dynamic MPVRPs | 107 |
| 4.4 | Characterization of PVRPs | 110 |
| 4.5 | Overview of solution approaches | 111 |
| | | |
| A.1 | Parameters for HALNS | 124 |
| A.2 | Detailed results for VRPTW instances | 126 |
| A.3 | Detailed Results for Belhaiza et al. (2014) VRPMTW Instances | 128 |
| A.4 | Detailed results for VRPSTW instances type 1 (only lateness), $\alpha = 1$ | 130 |
| A.5 | Detailed results for VRPSTW instances type 2 (earliness and lateness), $\alpha = 1$ | 131 |
| A.6 | Results for VRPAP instances | 132 |
| | | |
| B.1 | Parameters for HALNS | 135 |
| B.2 | Analysis of Runtime Scaling Factor | 136 |
| B.3 | CPUs used | 136 |
| B.4 | Detailed results on 2E-VRP instances | 137 |
| B.5 | Detailed results on LRP instances | 141 |
| B.6 | Detailed results on MDVRP instances | 143 |
| B.7 | Solution LRP instance <i>P122212</i> | 144 |
| B.8 | Solution LRP instance <i>P123212</i> | 145 |
| B.9 | Solution LRP instance <i>P123222</i> | 146 |

Contribution 1 - Optimizing Routing and Delivery Patterns with Multi-compartment Vehicles

1 Optimizing Routing and Delivery Patterns with Multi-compartment Vehicles

Markus Frank, Manuel Ostermeier, Andreas Holzapfel, Alexander Hübner, Heinrich Kuhn

Abstract Retailers usually apply repetitive weekly delivery patterns when scheduling the workforce for shelf replenishment, defining cyclic transportation routes and managing warehouse capacities. In doing so, all logistics subsystems are jointly scheduled. Grocery products require different temperature zones. As long as transport was in separated vehicles due to temperature requirements, it was not possible to coordinate deliveries across different temperature zones. The recent introduction of multi-compartment trucks has changed this and allows joint deliveries. This simultaneous delivery of multiple product segments impacts repetitive weekly delivery patterns as, for example, low volume segments can be delivered more frequently if they are transported together with high volume segments. We address the problem of defining delivery patterns for delivery with multi-compartment vehicles. After deriving decision-relevant costs, we propose a novel model that defines the Periodic Multi-Compartment Vehicle Routing Problem. The model is solved by an integrated framework that determines delivery patterns within an Adaptive Large Neighborhood Search in combination with a Large Neighborhood Search for solving the routing problem. We analyze the impact of selecting delivery patterns across product segments and show the efficiency of our integrated planning approach using numerical studies. Joint planning generates cost savings of up to 15%. Furthermore, we show that the algorithm provided can also improve single-segment problems by 3% compared to a state-of-the art benchmark. Beyond that we demonstrate the applicability and advantage of our approach in a case study with a large German grocery retailer.

Accepted in European Journal of Operational Research, Volume 293, Issue 2, Pages 495-510

URL: <https://doi.org/10.1016/j.ejor.2020.12.033>

1.1 Introduction & Motivation

Retailers constantly strive for excellence in logistics due to tight margins, heavy competition and high customer expectations. This may be achieved with new technologies and advanced planning approaches that enable better coordination within and between subsystems in a retail supply chain. Grocery stores exhibit a repetitive sales pattern. Consequently, grocers define store-specific delivery patterns (DP) and cyclically supply stores with their requirements of products and goods. The DPs constitute a defined combination of weekdays on which a store is supplied. They are usually defined as part of the tactical planning for standard weeks without external influences (e.g., public holidays) (Kuhn and Sternbeck 2013). Defining the DPs also means determining the delivery frequency. The delivery frequency, however, impacts the volume per store delivery, which in turn affects the associated logistics costs in the distribution center (DC), transportation and store. For example, while a larger delivery volume is beneficial for picking and transportation processes due to economies of scale, it is unfavorable for store processes as storage space within stores is usually very limited (Taube and Minner 2018). Deliveries that do not fit onto the shelves require extra handling and intermediate storage in the backroom (Kotzab and Teller 2005, Reiner et al. 2013). The resulting tradeoff for distribution, warehouse and store costs has to be considered within the planning process (Sternbeck and Kuhn 2014).

Furthermore, the distribution process in grocery retail involves multi-temperature logistics due to the different temperature requirements across products. These requirements range right from sub-zero temperatures for the transport of frozen products, lightly cooled products, through to ambient products without any temperature regulation. The exact temperatures are strictly regulated by law and the cooling chain must not be violated during the processing of orders. Retailers usually categorize product segments to different product groupings with similar temperature requirements (e.g., deep-frozen, fresh, dairy, ambient) and organize warehouses by temperature zones. These segments were generally distributed individually in the past. However, multi-compartment vehicles (MCVs) – recently introduced in a flexible version for food transportation – allow multi-temperature transportation. These trucks enable the joint transportation of multiple segments, i.e., products with differing temperature requirements on the same vehicle. The loading area of an MCV can be split flexibly into different compartments for each tour, and the temperature of each compartment can be adjusted individually (Ostermeier and Hübner 2018, Ostermeier et al. 2021). This allows for high flexibility in assigning orders to tours and sequencing the individual routes. MCVs also open up new possibilities for the definition of DPs. The DPs of different segments of a store can be aligned to achieve transportation synergies, which may result in higher delivery frequencies for the store. For instance, frozen products are often delivered once or twice per week due to small order volumes. If combined with other segments (e.g., fresh or ambient products), the frequency can be adjusted to enable more frequent deliveries. The simultaneous supply of different segments reduces the number of stops per route (Hübner and Ostermeier 2019) and increases the probability that the products delivered could be entirely stacked on the shelf, since greater delivery frequency will decrease the product volume per delivery (van Zelst et al. 2009, Donselaar et al. 2010, Reiner et al. 2013).

In current literature the definition of DPs differs for each product segment and generally assumes store deliveries with single-compartment vehicles (SCV) (e.g., Gaur and Fisher (2004), Sternbeck and Kuhn (2014), Holzapfel et al. (2016), Taube and Minner (2018)). The joint delivery of multiple segments is not considered. This raises the question of how the combination of the supply across multiple product segments influences the definition of store-specific DPs for individual segments, and how the altered DPs affect total logistics costs. To address this question, we formulate the Periodic Multi-Compartment Vehicle Routing Problem (PMCVRP), including the definition of DPs and decisions on the corresponding delivery schedules. To further detail the problem, we provide the related problem characteristics and literature in Sections 1.2 and 1.3. Section 1.4 presents the PMCVRP that considers several product segments demanding different temperature zones. The PMCVRP simultaneously decides on (i) the optimal delivery frequency and days for each segment and each store, and (ii) the optimal delivery of the associated store orders with multi-compartment vehicles. The decision model formulated explicitly takes into account the interdependency between delivery frequency and routing decisions. The resulting problem is NP-hard since it is a generalization of the capacitated VRP (Toth and Vigo 2014), and thus a heuristic solution approach is presented for practice-relevant problem sizes in Section 1.5. We introduce an approach that iteratively addresses the multi-period problem of defining DPs with an Adaptive Large Neighborhood Search (ALNS) and the corresponding routing problem with MCVs with a Large Neighborhood Search (LNS). To the best of our knowledge, this is the first comprehensive model and solution approach for this problem. Section 1.6 provides numerical studies, and Section 1.7 summarizes our findings and refers to future research opportunities.

1.2 Grocery Supply Chain, associated Processes and Costs

1.2.1 Distribution Processes in Grocery Retailing

Grocery retailers channel about 70% to 90% of shipment volumes to their stores via DCs. Most retailers operate their own vertically integrated logistics network with several central and regional DCs, a vehicle fleet and a large number of local stores to manage (Kuhn and Sternbeck 2013). Usually between 50 and 400 outlets are served from a single DC (Glatzel et al. 2012). In this context, the internal grocery retail supply chain can be divided into three logistics subsystems: DC, transportation and store. The store delivery process can be characterized as follows. The products of a store order are picked onto pallets or roll cages in the DC. Next, trucks transport the goods to the stores. Store employees then bring the load carriers to the show room and direct shelf filling takes place (Reiner et al. 2013). If products do not fit onto the shelves, the remaining units are carried to the backroom of the store. Refilling takes place later, when space becomes available due to consumer purchases (Kotzab and Teller 2005, Kuhn and Sternbeck 2013, Holzapfel et al. 2016).

Groceries are stored in and transported from DCs to stores in different temperature zones. The specific temperature requirements during storage and transportation are subject to legal regulations. In the European Union, temperatures of -20°C to -18°C for deep-frozen products, $+2^{\circ}\text{C}$ to $+7^{\circ}\text{C}$ for cooled products (like meat and dairy products), and $+4^{\circ}\text{C}$ to $+7^{\circ}\text{C}$ for fruits and vegetables are

mandatory. For some fresh products, retailers apply further product-specific temperature zones to obtain a longer shelf life (e.g., a maximum temperature of $+2^{\circ}\text{C}$ for fresh fish and seafood). Only ambient products like dry goods and beverages do not need to adhere to specific transportation temperature requirements. Considering the mandated temperature zones plus ambient products, there are at least four different zones in grocery distribution. On the grounds of temperature requirements, retailers store, pick, and prepare the deliveries in temperature-specific DC areas. The traditional approach is to distribute goods separately for each product segment with their specific temperature requirements. Recent truck models are equipped with temperature-specific compartments that allow the transport of different product segments in the corresponding chambers (compartments) of one truck (Ostermeier and Hübner 2018). For example, when considering deep-frozen and ambient products ordered by the same outlet, the use of such MCVs makes it possible to deliver both product segments on the same truck at the same time. Whereas the transport of the different product segments needs to be planned separately when SCVs with one temperature zone are applied, it becomes necessary to jointly plan flows across segments when MCVs are available. The loading area of an MCV is customized for each tour. Each compartment can be adjusted to a given temperature according to the requirements of loaded product segments. The delivery process with MCVs starts with the collection of orders for all segments assigned to the corresponding tour. Collection involves the approach of multiple shipping gates as each segment is stored in a separate area at the DC (see Figure 1.1). After all segments are loaded, the MCV jointly supplies the corresponding stores with the different product segments. Figure 1.1 illustrates the overall process of an MCV tour with four segments.

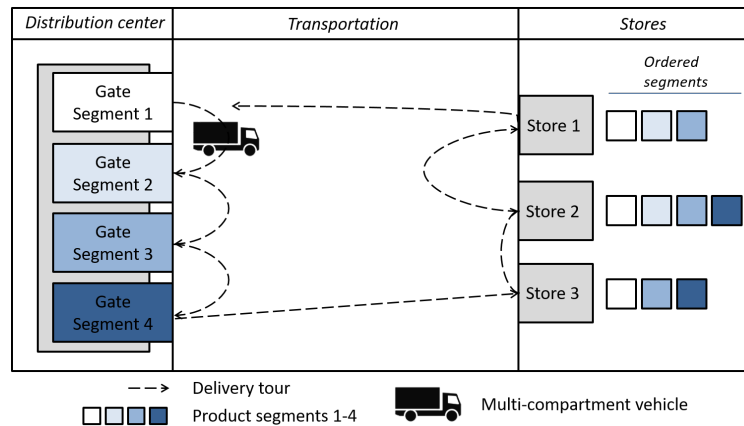


Figure 1.1: Distribution process with MCVs

1.2.2 Selection of Delivery Patterns

Theoretically, each store could be supplied individually whenever an order is triggered. However, retailers limit the delivery frequency to a certain degree for practical reasons and use weekly delivery cycles. Applying such repetitive and store- and segment-specific DPs has several reasons. Retailers usually apply periodic inventory review policies (see e.g., Broekmeulen et al. (2006), Curşeu et al. (2009), Minner and Transchel (2010)). A replenishment order is issued whenever shelf inventory

falls to or below a reorder level. Such a cyclic ordering policy allows order volumes to be adapted and eases subsequent logistics planning. The orders arrive at a store on identical weekdays each week. Scheduling the workforce for the shelf replenishment process is therefore much easier. Likewise, in terms of transportation, such cyclic ordering and defined delivery days offer the opportunity to design cyclic master routes for each week. At the DC, shift planning can be adjusted with regard to expected picking volumes that are dependent on the delivery frequency determined across all stores (Holzapfel et al. 2016). Finally, retail practice considers the selection of DPs as an important lever to balance DC, transportation and instore requirements (Hübner et al. 2013, Sternbeck and Kuhn 2014).

Assuming a one-week delivery cycle with six delivery days allows for one to six deliveries per week and store, resulting in $2^6 - 1 = 63$ possible DPs. This yields $63^{|N|}$ possible combinations for $|N|$ stores. The majority of retailers apply store- and segment-specific DPs. This means that individual patterns for each product segment are defined for each store. This is motivated by the fact that both stores (regarding sales volumes and shelf capacity) and segments (regarding freshness requirements) are heterogeneous. As such, the combinatorial challenge increases to $63^{|N| \times |S|}$ possibilities, where $|S|$ indicates the number of product segments. An example for store- and segment-specific DPs is shown in Figure 1.2. It illustrates three different DPs for the corresponding segments of a single store.

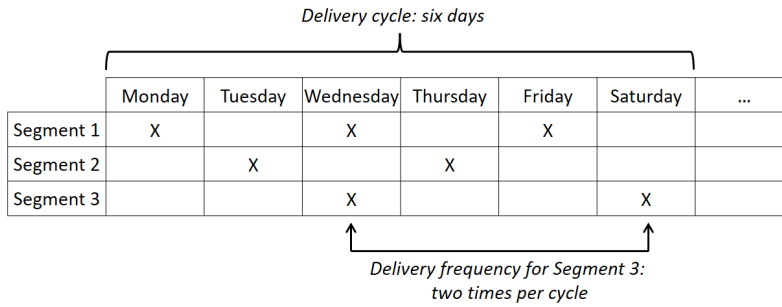


Figure 1.2: Example of delivery patterns for a store with three segments

1.2.3 Identification of decision-relevant Costs and Constraints for defining DPs and using MCVs

A DP defines the delivery frequency (e.g., three times per week) and the corresponding delivery days (e.g., each Monday, Wednesday and Friday). It therefore also determines the delivery quantities for each day selected. The daily demand of each store for each segment can be estimated and builds the foundation of the planning. When using daily demand, the weekly seasonality is also incorporated (e.g., higher demand on Saturdays). The demand between order intervals is aggregated to the preceding delivery, e.g., if a delivery happens on Monday and Wednesday, the demand for Tuesday will be fulfilled on Monday. This means that the delivery size of each segment and each store in each period results from selecting a DP. A DP with a higher weekly delivery frequency leads to more but smaller deliveries while the total delivery quantity remains constant. Hence, as order intervals

are a result of the DPs applied, volume effects along the supply chain occur that strongly influence operations and costs along the whole grocery supply chain. When applying MCVs, delivery days can be optimized across segments. This also impacts the frequency and size of deliveries of the segments.

The following details the processes, constraints and costs involved in the corresponding subsystems. The analysis is based on our work with a case company and related literature, in particular Sternbeck and Kuhn (2014), Holzapfel et al. (2016) and Hübner and Ostermeier (2019).

Distribution center The store orders are processed in segment-specific areas of the DC. These areas particularly fulfill the specific temperature requirements of the individual products of the segments. The picking volume in each of these areas is generally limited on each working day of the week. In addition, retailers define minimum workload levels for each area to balance the workloads between consecutive working days. This avoids workload peaks and eases shift scheduling.

Furthermore, each order causes *order processing costs* at the associated area of the DC. These are fixed costs for each order. A higher delivery frequency therefore leads to an increasing number of orders and higher overall order processing costs at the DC. After the order processing, the respective products are picked and then placed on a load carrier. The entire order is finally packaged and placed at the DC area's gate for loading and transportation. These are variable costs that depend on the delivery size. A higher delivery frequency leads to smaller pick sizes and thus to higher overall *picking and packing costs*.

Transportation Subsequent to the picking and packing in the DC, the product segments are loaded onto trucks with limited capacity. The associated costs are denoted as *loading costs*. As described above, the distribution process with MCVs requires the collection of segments from different DC areas. These costs depend on the number of segments assigned to a tour and thus on the number of compartments required on the vehicle. For the actual delivery tour, transportation costs arise that involve costs for traveling between the locations and costs for unloading the goods at the stores. The *travel costs* depend on the distance covered by a truck between the locations, i.e., DCs and stores. The transportation costs obviously increase with higher frequency and more tours, but may also stagnate if segments are transported jointly across segments. *Unloading costs* occur when a truck stops at a store and unloads the delivery. These costs are induced by setup times for unloading goods from the vehicles and goods receiving processes in the store. The latter include tasks of store employees for checking the items received and complete administrative steps for the goods reception. The resulting costs are fixed costs for each receiving process. The entire unloading costs can therefore be reduced if DPs are synchronized across multiple segments, since this will reduce the number of stops required at the stores.

Store At each arrival of a new delivery at a store, the orders are further processed. The entire store receiving capacity is generally limited because of space and workforce limitations. This limits the entire volume that can be delivered on a single day. A delivery is either immediately used for direct shelf filling purposes or stored in a backroom until refilling is required. The associated

refilling costs are independent of the usage of MCVs. The following costs only depend on the delivery frequency and size. First of all, *direct shelf filling costs* represent the transport of goods received from the store inbound area to the shelves and putting the units onto the shelves. They are store- and segment-specific and depend on the different settings concerning store layout and shelf types. If the quantity of a product delivered exceeds the shelf capacity, the remaining units have to be brought to the backroom and are stored there until the required capacity is available due to customer purchases and a refill can take place. The related costs are denoted as *shelf refilling costs* from the backroom. This additional refill process is considerably more costly than direct shelf filling. With a higher frequency of deliveries and smaller delivery sizes, this process becomes less frequent and less capacity may be required to store additional units in the backroom that did not fit on the shelves. Finally, new orders are submitted as soon as the reorder level is reached. The reorder level depends on the order size and therefore on the delivery frequency. Each order causes fixed *order placement costs* at the stores. These costs increase with higher delivery frequency.

Summary Some of the costs above depend on the same decisions and can therefore be summarized in a single cost parameter to streamline the cost model. First of all, a pattern-dependent cost parameter is introduced that comprises costs for order processing and picking and packing at the DC as well as direct shelf filling, shelf refilling, and order placement costs at the store. The DP defines – via the chosen delivery days – the total number of deliveries per period, e.g., per week, and the corresponding delivery sizes of each segment per store delivery, which affect all the cost factors mentioned. Secondly, travel and unloading costs can be summarized, as they both depend on the routing. Figure 1.3 summarizes the decision-relevant costs and constraints per subsystem considered in the present paper. In addition it lists the relevant constraints that have to be taken into account.




| Subsystem | Distribution Center | | Transportation | | | Store | | |
|-----------------------------|---|-------------------------|---|----------------------|-----------------|---|-----------------------|-----------------------|
| |  | |  | | |  | | |
| Relevant Costs | Order processing costs | Picking & packing costs | Loading costs | Travel costs | Unloading costs | Direct shelf filling costs | Shelf refilling costs | Order placement costs |
| Impacting Variables | Delivery pattern | | Number of compartments | Routing | | Delivery pattern | | |
| Aggregated Costs | Pattern-dependent costs | | Loading costs | Transportation costs | | Pattern-dependent costs | | |
| Relevant Constraints | Minimum & maximum picking capacity | | Vehicle capacity | | | Store receiving capacity | | |

Figure 1.3: Decision-relevant operations, distribution costs and constraints along the internal retail supply chain

1.3 Related Literature

The problem considered generally belongs to the class of periodic vehicle routing problems (PVRP). There is a wide range of publications available concerning the PVRP (e.g. Campbell and Wilson (2014)). Classical PVRP literature however neglects several essential characteristics that are relevant when planning DP and MCV in grocery retailing. We therefore focus our literature review on publications related to DP and MCV planning.

Literature on DP planning Publications on DP planning consider – contrary to pure PVRP publications – pattern-dependent costs, and analyze their influences on overall planning. They especially take into account that the delivery sizes per day depend on the DPs chosen. An approach to determine a weekly delivery schedule is provided by Gaur and Fisher (2004) based on a periodic inventory routing problem. Ronen and Goodhart (2008) consider a related problem and include DC costs and additional extensions, such as limited picking capacity, a heterogeneous fleet, and daily minimum utilization rates for DC and transportation subsystems. Similar stores are clustered and patterns are predefined for these clusters using an MIP. Furthermore, a PVRP is applied for the routing. Clustering, pattern-definition and routing is not done sequentially without any feedback loops. This causes the problem that once patterns are assigned to the stores they cannot be changed any more although the routing step may reveal that adapting the delivery patterns could capture additional savings (Holzapfel et al. 2016). In addition, they neglect instore operational costs. Sternbeck and Kuhn (2014) are the first to examine the logistics processes comprehensively in DCs, transportation and stores and their dependencies on DPs. They develop a binary integer program that minimizes the sum of all relevant costs identified and apply it to a real-life case. Transportation costs are approximated with a cost matrix dependent on distance and order size. Actual tours are not considered. Holzapfel et al. (2016) also take into account DC, transportation and instore logistics and propose an advanced solution approach that clusters stores and approximates transportation costs using the logic of Fisher and Jaikumar (1981). Taube and Minner (2018) focus on handling costs at the DC and stores. They consider a classical joint replenishment problem with stochastic demand and present decomposition approaches and a genetic algorithm to solve it. After experimenting with random data instances, they use the most promising model for a case study with a European retailer.

To sum up, this literature stream is related to our problem setting, but falls short in approximating transportation costs without directly solving the related VRP or neglecting instore costs. Furthermore, MCVs and product flows across segments have not been investigated so far.

Literature on the MCVRP The largest body of MCVRP literature deals with applications in fuel distribution and fixed compartment sizes (e.g., Avella et al. (2004), Coelho and Laporte (2015)). Yet in our problem context, the flexibility of compartments is a central characteristic and we therefore focus on related publications. The first comprehensive formulation of a vehicle routing problem with both fixed and flexible compartments is presented by Derigs et al. (2011). The authors use and evaluate a whole string of heuristic solution methods (construction-, search- and metaheuristics) for

the problem, while focusing on their application in food and petrol distribution. Henke et al. (2015) discuss an MCVRP with flexible compartments for application in German glass waste collection. A VNS is used to improve an initial solution generated by a randomized construction procedure. Koch et al. (2016) and Henke et al. (2019) consider a similar problem formulation but propose different solution approaches. Koch et al. (2016) present a genetic algorithm that may also be modified for a multi-period context and Henke et al. (2019) develop a branch-and-cut algorithm to address the problem. Hübner and Ostermeier (2019) consider an MCVRP in the context of grocery retailing, taking into account MCV-specific costs for the first time. An LNS is applied to solve the corresponding problem. Ostermeier and Hübner (2018) also extend this research and present a vehicle selection model for the MCVRP. Furthermore, Ostermeier et al. (2018) consider the use of flexible compartments and corresponding loading issues. The authors present a mathematical formulation for the extended MCVRP and solve the problem with a branch-and-cut approach as well as an adapted LNS. Besides flexible compartments, Hsiao et al. (2017) also consider the flexible adjustment of compartment temperatures and present a biogeography-based optimization approach. Martins et al. (2019) present an MCVRP for multiple periods that considers consistent deliveries across segments but uses the given DPs as input parameter. They solve the resulting multi-period MCVRP with product-oriented time windows using an ALNS. For a review on MCVRP literature, we further refer to Ostermeier et al. (2021).

To sum up this literature stream, we can state that current MCVRP literature for flexible compartments has been developed only recently. Multi-period problems are rare and – if available – do not consider the assignment of DPs and, particularly, the choice of delivery days.

Summary Despite the numerous publications on DPs and MCVs, none of the above integrates the diverse problem characteristics mentioned. We address this gap in literature and present a problem formulation that considers the joint selection of DPs across stores and segments in the circumstances of MCV deliveries. Moreover, the interrelation of the joint DP selection of product segments on warehousing, MCV routing, and store operations requires an integrative planning approach. Only the simultaneous consideration of all decision-relevant costs and constraints ensures feasible and cost-optimal decisions for the entire distribution process. As such, our work extends the literature on both DP and MCV planning in grocery distribution. Our research specifically makes a contribution to:

- Identifying decision-relevant costs in warehousing, transportation and instore operations when selecting DPs across product segments and using MCVs for their joint store deliveries
- Formulating a novel model, i.e., the PMCVRP that simultaneously defines cost-minimal DPs and MCV delivery tours for a diverse set of product segments in a multiple period environment
- Developing a sophisticated heuristic solution algorithm that finds good solutions in acceptable computation times for the defined PMCVRP model, and
- Generating numerical examples with simulated and actual retail data to obtain insights into the value of integration when DPs of products with different temperature requirements are jointly delivered from the DCs to the stores.

1.4 Decision Model for Periodic Multi-compartment Vehicle Routing Problem

In the present section we formulate the mathematical model of the decision problem described. The model is based on the PVRP where transportation is executed by MCVs, and DC-, transportation and store-related costs are considered that depend on the DP chosen. We denote this problem as PMCVRP. The model is formulated as follows using the notation given in Table 1.1.

| Sets | |
|----------------------------------|--|
| K | Set of vehicles $K = \{1, \dots, K \}$ |
| N | Set of stores $N = \{1, \dots, N \}$, $N_0 = \{0, 1, \dots, N \}$ with 0 as depot |
| P | Set of delivery patterns $P = \{1, \dots, P \}$ |
| S | Set of product segments $S = \{1, \dots, S \}$ |
| T | Set of periods $T = \{1, \dots, T \}$ |
| Parameters | |
| a_{pt} | $a_{pt} = 1$, if period t , $t \in T$, is included in pattern p , $p \in P$, 0 otherwise |
| c_s^{load} | Loading costs for segment s , $s \in S$ |
| c_{ij}^{tran} | Transportation costs for approaching location j after i for $i, j \in N_0$, where $c_{ij}^{\text{tran}} = c_{ij}^{\text{travel}} + c_j^{\text{unload}}$ |
| c_{psi}^{pat} | Pattern-dependent costs of segment s , $s \in S$, and store i , $i \in N$, when pattern p , $p \in P$, is selected |
| o_{psit} | Delivery quantity of segment s , $s \in S$, for store i , $i \in N$, in period t , $t \in T$, when pattern p , $p \in P$, is selected |
| Q^{veh} | Vehicle capacity |
| Q_s^{pickmin} | Minimum picking capacity for segment s , $s \in S$, at the DC |
| Q_s^{pickmax} | Maximum picking capacity for segment s , $s \in S$, at the DC |
| Q_i^{recmax} | Maximum receiving capacity for store i , $i \in N$ |
| Decision and auxiliary variables | |
| u_{skt} | Binary; indicating whether segment s , $s \in S$, is delivered by vehicle k , $k \in K$, in period t , $t \in T$ |
| x_{ijkt} | Binary; indicating whether vehicle k , $k \in K$, travels from location i to j for $i, j \in N_0$, in period t , $t \in T$ |
| y_{sikt} | Binary; indicating whether store i , $i \in N$, receives segment s , $s \in S$, in period t , $t \in T$, by vehicle k , $k \in K$ |
| z_{psi} | Binary; indicating whether pattern p , $p \in P$, is selected for segment s , $s \in S$, and store i , $i \in N$ |

Table 1.1: Notation used to model PMCVRP

Let $G = (N_0, E)$ be an undirected, weighted graph consisting of a vertex set $N_0 = \{0, 1, \dots, |N|\}$, representing the location of the depot (0) and the locations of stores ($N = \{1, \dots, |N|\}$), and a set of edges $E = \{(i, j) : i, j \in N_0\}$, representing the connection between different locations. Each edge is associated with non-negative travel costs c_{ij}^{travel} . The product segments are denoted by the set $S = \{1, \dots, |S|\}$. The stores are supplied using a heterogeneous fleet of MCVs denoted by the set of vehicles $K = \{1, \dots, |K|\}$. The vehicle fleet is assumed to be sufficiently large to satisfy the total demand. The compartment setting for each MCV is adjustable, i.e., the number and size of

compartments is not predetermined but part of the decision problem. Further, the total vehicle capacity Q^{veh} is not affected by the specific compartment setting due to the given flexibility. Each MCV is used for one tour per period at most. The planning horizon comprises one delivery cycle with a given set of delivery periods, $T = \{1, \dots, |T|\}$. Further, a set of possible delivery patterns $P = \{1, \dots, |P|\}$ is introduced that covers all possible store- and segment-specific delivery schedules. Generally, this set can include all feasible weekday combinations of all frequencies, but a prior limitation, e.g., dependent on certain segment or store features, is reasonable.

Every store has a positive demand for each segment across the planning horizon. The delivery quantity o_{psit} indicates the demand of store i for product segment s at period t when DP p is selected. It depends on the pattern assigned, and is part of the decision problem. The total demand of a segment is split across actual days of delivery according to the chosen DP. The quantity of a delivery must include the demand of the period during which the delivery takes place and the demand of all following periods until the next scheduled delivery. In line with this, the parameter a_{pt} indicates whether period t is included in pattern p or not. Moreover, single orders for one segment in one period may not be split up across different vehicles.

The cost parameters are defined as follows. The loading costs c_s^{load} represent the costs for stopping at a segment-specific gate at the depot and for loading the order onto the truck. Travel costs c_{ij}^{travel} include the costs for the travel from location i to location j . Unloading costs c_j^{unload} cover the costs for each stop at a store. For the sake of simplicity, we summarize travel and unloading costs in a generalized cost term for transportation, and define $c_{ij}^{\text{tran}} := c_{ij}^{\text{travel}} + c_j^{\text{unload}}$. Finally, the pattern-dependent costs c_{psi}^{pat} indicate the costs that occur when store i is supplied with segment s according to pattern p . The pattern-dependent costs comprise both depot- and store-specific handling costs as described in Section 1.2. The following binary decision variables are applied:

- x_{ijkt} indicates whether vehicle k travels from store i to j within period t , $k \in K, i, j \in N_0, t \in T$.
- y_{sikt} indicates whether store i receives segment s by vehicle k within period t , $s \in S, i \in N, k \in K, t \in T$.
- z_{psi} indicates whether pattern p is selected for segment s and store i , $p \in P, s \in S, i \in N$.

Additionally, we introduce the auxiliary binary variables u_{skt} indicating if vehicle k contains at least one order of segment s on day t . The mathematical model for the PMCVRP can then be formulated as follows.

$$\min \quad \text{TC} = \sum_{s \in S} \sum_{k \in K} \sum_{t \in T} c_s^{\text{load}} \cdot u_{skt} + \sum_{i \in N_0} \sum_{\substack{j \in N_0 \\ i \neq j}} \sum_{k \in K} \sum_{t \in T} c_{ij}^{\text{tran}} \cdot x_{ijkt} + \sum_{p \in P} \sum_{s \in S} \sum_{i \in N} c_{psi}^{\text{pat}} \cdot z_{psi} \quad (1.1)$$

subject to

$$\sum_{p \in P} z_{psi} = 1 \quad \forall s \in S, \forall i \in N \quad (1.2)$$

1 Optimizing Routing and Delivery Patterns with MCVs

$$\sum_{\substack{i \in N_0 \\ i \neq j}} x_{ijkt} = \sum_{\substack{i \in N_0 \\ i \neq j}} x_{jikt} \quad \forall k \in K, \forall t \in T, \forall j \in N_0 \quad (1.3)$$

$$\sum_{k \in K} \sum_{j \in N_0} x_{0jkt} \leq |K| \quad \forall t \in T \quad (1.4)$$

$$\sum_{s \in S} y_{sjkt} \leq |S| \cdot \sum_{i \in N_0} x_{ijkt} \quad \forall j \in N, \forall t \in T, \forall k \in K \quad (1.5)$$

$$\sum_{i \in L} \sum_{j \in L} x_{ijkt} \leq |L| - 1 \quad \forall t \in T, \forall k \in K, \forall L \subseteq N, |L| \geq 2 \quad (1.6)$$

$$\sum_{j \in N_0} x_{0jkt} \leq 1 \quad \forall t \in T, \forall k \in K \quad (1.7)$$

$$\sum_{p \in P} \sum_{s \in S} \sum_{i \in N} o_{psit} \cdot z_{psi} \cdot y_{sikt} \leq Q^{\text{veh}} \quad \forall t \in T, \forall k \in K \quad (1.8)$$

$$Q_s^{\text{pickmin}} \leq \sum_{p \in P} \sum_{i \in N} o_{psit} \cdot z_{psi} \leq Q_s^{\text{pickmax}} \quad \forall s \in S, \forall t \in T \quad (1.9)$$

$$\sum_{p \in P} \sum_{s \in S} o_{psit} \cdot z_{psi} \leq Q_i^{\text{recmax}} \quad \forall i \in N, \forall t \in T \quad (1.10)$$

$$\sum_{k \in K} y_{sikt} = \sum_{p \in P} z_{psi} \cdot a_{pt} \quad \forall s \in S, \forall i \in N, \forall t \in T \quad (1.11)$$

$$\sum_{i \in N} y_{sikt} \leq u_{skt} \cdot |N| \quad \forall s \in S, \forall k \in K, \forall t \in T \quad (1.12)$$

$$u_{skt} \in \{0, 1\} \quad \forall s \in S, \forall k \in K, \forall t \in T \quad (1.13)$$

$$x_{ijkt} \in \{0, 1\} \quad \forall i, j \in N_0, \forall k \in K, \forall t \in T \quad (1.14)$$

$$y_{sikt} \in \{0, 1\} \quad \forall s \in S, \forall i \in N, \forall k \in K, \forall t \in T \quad (1.15)$$

$$z_{psi} \in \{0, 1\} \quad \forall p \in P, \forall s \in S, \forall i \in N \quad (1.16)$$

The objective function (1.1) minimizes the total costs (TC), consisting of loading, transportation (including unloading), and pattern-dependent costs that arise for every pattern that is assigned to a segment-store combination (s, i) , $s \in S, i \in N$. Constraints (1.2) ensure that exactly one delivery pattern per product segment is assigned for each store. Constraints (1.3) represent the flow conservation, guaranteeing that every store visited is also left again. Additionally, each vehicle has to start from the depot as defined by Constraints (1.4). Constraints (1.5) guarantee that a store is visited if a corresponding order is loaded. The subtour elimination constraints are denoted by Constraints (1.6). According to Constraints (1.7), every vehicle may be used only once per day. Constraints (1.8) ensure that the vehicle capacity is not exceeded. Constraints (1.9) ensure that the picking effort at each segment-specific DC area neither falls below the minimum nor exceeds the maximum picking capacity on each day t . Constraints (1.10) consider the maximum receiving capacity of each store i on each delivery day t . Constraints (1.11) ensure that if a store receives a product segment on day t according to the selected DP, the corresponding segment has to be assigned to a vehicle on this delivery day. Further, if at least one order of segment s is assigned to vehicle k , the corresponding compartment is required and thus, u_{skt} is activated (Constraints (1.12)). Lastly, the decision and auxiliary variables are defined as binary by Constraints (1.13)–(1.16).

The PMCVRP extends both PVRP and MCVRP. As such, it generalizes the well-known CVRP that is known to be an NP-hard optimization problem (see e.g., Laporte (2009), Toth and Vigo (2014)). Exact solution approaches are only able to solve small problem instances. In our application we consider industry cases with hundreds of stores that are served from temperature-specific DCs with a diverse set of product segments. In these cases, heuristics are required to provide solutions for the PMCVRP.

1.5 Solution Approach

We propose a heuristic algorithm to solve the PMCVRP. The algorithm iteratively optimizes the assignment of DPs for each segment-store combination and solves the corresponding MCVRP in each period of the planning horizon. Figure 1.4 illustrates the general framework of the algorithm proposed. It contains three major parts that are described in more detail within the upcoming section. After generating an initial solution (see Section 1.5.1), the algorithm performs two sequential stages within its second part (see Section 1.5.2). In Stage 1, an ALNS framework is used to determine individual DPs for each segment-store combination (see Section 1.5.2.1). This results in new partial solutions that define the delivery quantities for each period and each segment-store combination of the entire planning horizon. Stage 2 then solves the resulting MCVRPs applying an LNS approach in each period of the planning horizon (see Section 1.5.2.2). The ALNS optimizes delivery patterns across all periods, whereas the LNS only optimizes the routing within a period. Finally a Simulated Annealing approach is used to decide on the next candidate schedule to work on during the subsequent iteration. This part of the algorithm also adapts the parameters of the search process (Section 1.5.3).

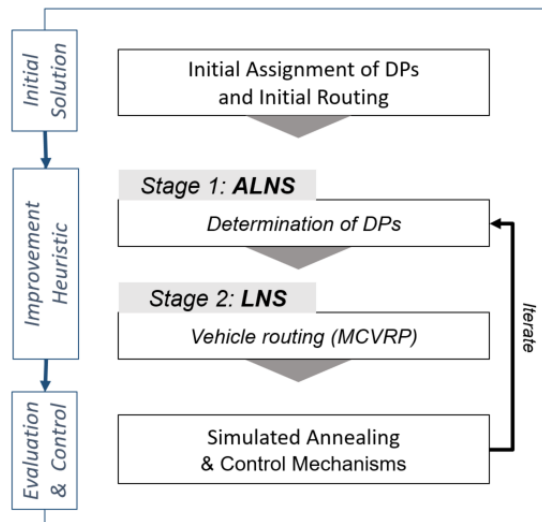


Figure 1.4: Algorithmic structure

Please note that we use the following terminology within the detailed description of the algorithm. Deliveries are set by segment and store and are therefore uniquely defined for each segment-store combination subject to the assigned patterns. We will therefore use the term “segment-store com-

bination" (s, i) , $s \in S, i \in N$ to uniquely define the object and planning entity. We use this term whenever we consider the characteristics attributed to a store and the corresponding segment (e.g., weekly demand for segment s by store i). Furthermore, a DP p is assigned to each segment-store combination, indicated by the triple (p, s, i) , $p \in P, s \in S, i \in N$.

1.5.1 Initial Solution

The solution approach starts with a random initial assignment of patterns. This assignment specifies the delivery days and the associated delivery quantities for each segment-store combination. It also determines the entire delivery requirements in each period. Afterwards, the procedure applies the Savings Algorithm by Clarke and Wright (1964) to construct feasible delivery tours in each period. The Savings Algorithm was used in many VRP and MCVRP formulations and provides fast and reasonable initial solutions (Toth and Vigo 2014). We use the parallel version of the algorithm as it provides better solutions than the sequential approach (Laporte 2009). The procedure starts constructing single tours for every segment-store delivery scheduled in each period of the planning horizon. Afterwards we calculate the associated savings values for all pairs of segment-store combinations $[(s, i), (\tilde{s}, j)]$, $s, \tilde{s} \in S, i, j \in N$, if they were jointly delivered: $\Delta c_{si, \tilde{s}j} = c_{i0}^{\text{travel}} + c_{0j}^{\text{travel}} - c_{ij}^{\text{travel}}$. Here, 0 represents the depot and c_{ij}^{travel} denotes the travel costs between the respective locations i and j ($i, j \in N$). Iterating across the sorted decreasing list of savings-values $\Delta c_{si, \tilde{s}j}$, the corresponding tours of the pairs (s, i) and (\tilde{s}, j) are merged if feasible.

1.5.2 Improvement Heuristic

In each iteration of our solution approach we change large parts of the current solution by assigning new delivery patterns to segment-store combinations (*Stage 1*) and solving the corresponding routing problems (*Stage 2*). Please note that we are keeping track of all partial solutions created (*Stage 1*), whereas the routing for each period (*Stage 2*) is only executed if a new solution (i.e., assignment of patterns) has been reached. In addition, we verify the feasibility of each pattern assignment in respect of the lower and upper picking limits at the DC and the store's maximum receiving capacity. In the event that it is not feasible, the partial solution is rejected and *Stage 1* is repeated. In the following both stages are described in detail.

1.5.2.1 Stage 1: ALNS for Optimizing DPs

The ALNS approach introduced by Shaw (1997) was effectively applied when solving multiple variants of the VRP (e.g., Shaw (1997), Ropke and Pisinger (2006)), and it was particularly effective when solving PVRPs (e.g., Zajac (2017)) and MCVRPs (e.g., Martins et al. (2019)). On the basis of an initial solution, it typically uses several remove and insertion operators to destroy and repair large parts of the solution in each iteration. In order to adapt the ALNS approach to the present problem and to the state of the search process, a weight is assigned to each operator that determines how often it is selected during the search process. The weights are adjusted dynamically depending on the past performance of the respective operator with respect to the overall solution. We describe this adjustment procedure in detail in Section 1.5.3. We use the ALNS as it enables us to embed

a whole set of operators derived from and built with problem-specific knowledge. Further, the adaptive mechanism decides which operators to use for which type of problem instance.

Most ALNS approaches include remove and reinsertion operators for customers or orders to recreate large parts of the solution. In this aspect, our ALNS differs from other formulations. Rather than removing and reinserting orders or deliveries, the operators used in our approach select new patterns for segment-store combinations, i.e., they decide on how often and on which days a store receives the respective segments. Traditional remove and insert operators are not applicable for our problem as these usually assume that the order sizes per customer and period are independent of the solution. In our case, however, the order size per store delivery depends on the chosen DP. For example, modifying a current DP will at least omit, add or move one delivery day. Consequently, this will change the delivery size and the related costs of the associated deliveries since we assume a pre-defined weekly demand pattern for all segment-store combinations. Modifying an individual pattern of a segment-store combination may therefore result in a completely new delivery schedule. Table 1.2 summarizes the ALNS operators applied. It comprises the following operator types: (i) score-related, (ii) cost-related, (iii) move-related and (iv) random. Each operator will change the patterns in an iteration for a given number of segment-store combinations.

| Operator h | Operator type | Operator name |
|--------------|---------------|---------------------------------|
| 1 | (i) | Proximity Operator |
| 2 | | Segment Bundle Operator |
| 3 | | Sales Volume Operator |
| 4 | (ii) | Cost-related |
| | | Pattern-dependent Cost Operator |
| 5 | (iii) | Move-One Operator |
| 6 | | Move-Two Operator |
| 7 | (iv) | Random |
| | | Random Operator |

Table 1.2: Operators used within the ALNS algorithm

(i) Score-related operators The structure of our score-related operators is based on the well-known Removal Operator by Shaw (Shaw 1997). For each score-related operator we define a relatedness measure $R_{si,\tilde{s}j}$ for two segment-store combinations (s, i) and (\tilde{s}, j) , $i, j \in N$, $s, \tilde{s} \in S$. The algorithmic structure of the score-related operators is identical despite the different relatedness measures. We therefore present this general five-step structure first and then detail the individual operators. The general structure of the score-related operators is further given in Algorithm 1.

After a random segment-store combination (s, i) has been selected in *Step 1*, the relatedness $R_{si,\tilde{s}j}$ between delivery schedules for segment-store combination (s, i) and all other combinations (\tilde{s}, j) , $\tilde{s} \in S, j \in N$, is calculated in *Step 2* and ranked in ascending order according to the relatedness measure calculated, $R_{si,\tilde{s}j}$. The more related the attributes of two segment-store combinations, the more likely it is to obtain synergies in a joint consideration and the higher the expected additional cost savings by aligning the respective patterns. Subsequent to the relatedness calculated, an additional parameter α is used that determines the degree of randomization of the search. More precisely, after sorting all segment-store combinations (\tilde{s}, j) according to their score, a random number ζ , $\zeta \in [0, 1]$ is drawn in *Step 3*, selecting the combination that lays ζ^α down the ranking.

If the combination selected (\tilde{s}, j) has a different pattern p' , $p' \in P$ compared to the combination (s, i) , a new pattern is assigned to combination (\tilde{s}, j) in *Step 4*. Here, the new pattern is chosen randomly among all patterns \tilde{p} , $\tilde{p} \in P$ that have a higher pattern similarity $(\omega_{psi, \tilde{p}\tilde{s}j})$ than the previously assigned pattern. The pattern similarity is calculated using Equation (1.17). This metric is determined by the ratio of matching periods $\eta_{psi, \tilde{p}\tilde{s}j}^t$ for pattern p and \tilde{p} , to the total number of periods $|T|$. Matching periods are days where both patterns intend to carry out a store delivery.

$$\omega_{psi, \tilde{p}\tilde{s}j} = \frac{\sum_{t=1}^{|T|} \eta_{psi, \tilde{p}\tilde{s}j}^t}{|T|} \quad (1.17)$$

This process is repeated until the patterns for c segment-store combinations are adjusted. The resulting new solution \mathbb{S}^* is then the input for solving the MCVRPs in *Stage 2*.

Algorithm 1: Score-related operators

Input: Solution \mathbb{S} , set of patterns P , number of segment-store combinations to be changed c , degree of randomization α

Output: new partial solution \mathbb{S}^* with the updated pattern assignments for all segment-store combinations considered (s, i) , $s \in S, i \in N$

- 1 List $L = \emptyset$
- 2 randomly select a segment-store combination (s, i) , $s \in S, i \in N$ and add it to list L
- 3 **while** $|L| < c$ **do**
- 4 *Step 1:* randomly select a segment-store combination from list L , $(s, i) \in L, s \in S, i \in N$
- 5 *Step 2:* compute score $R_{si, \tilde{s}j}$ for all segment-store combinations $(\tilde{s}, j) \notin L, \tilde{s} \in S, j \in N$, and sort by their score in ascending order
- 6 *Step 3:* draw a random number $\zeta \in [0, 1]$ and select the segment-store combination (\tilde{s}, j) , $\tilde{s} \in S, j \in N$, that lays ζ^α down the ranking
- 7 **if** pattern p assigned to (s, i) differs from pattern p' assigned to (\tilde{s}, j) (i.e., $p \neq p', p, p' \in P$), **then**
- 8 *Step 4:* assign new pattern to segment-store combination (\tilde{s}, j) that is selected randomly among all patterns \tilde{p} , $\tilde{p} \in P$, with a higher pattern similarity $\omega_{psi, \tilde{p}\tilde{s}j}$ to the current pattern p of (s, i)
- 9 **else**
- 10 **continue;**
- 11 *Step 5:* add (\tilde{s}, j) to L

Proximity Operator The first score-related operator is based on the idea that it is usually cost-efficient to serve stores in geographical proximity using the same vehicle. In order to enable a conjoint delivery in each period of the planning horizon, the patterns of these segment-store combinations should be as similar as possible. If two neighboring stores and their segments or two segments of the same store share the same delivery days (but only if this is the case), they should be placed on the same tours by the subsequent routing decision and transportation synergies can be realized. To do this, the Proximity Operator tries to assimilate the patterns of segment-store combinations by selecting combinations (s, i) and (\tilde{s}, j) with a lower value $R_{si, \tilde{s}j}^1$ subject to the patterns p and \tilde{p} currently selected.

$$R_{si, \tilde{s}j}^1 = \beta \cdot \frac{c_{ij}^{\text{travel}}}{\bar{c}^{\text{travel}}} + (1 - \beta) \cdot \omega_{psi, \tilde{p}\tilde{s}j} \quad (1.18)$$

$R_{si, \tilde{s}j}^1$ comprises two components that are weighted by β : the geographical proximity of the respective stores i and j and the current similarity of their patterns p and \tilde{p} . The first metric

is expressed by the travel costs c_{ij}^{travel} between the store locations of i and j and the maximum travel costs between any two store combinations \bar{c}^{travel} . Second, the pattern similarity $\omega_{psi, \tilde{p}\tilde{s}j}$ (see Equation (1.17)) of the current patterns of the segment-store combinations considered is included as we aim at changing combinations with dissimilar patterns.

Segment Bundle Operator A further operator bundles deliveries across segments. Scheduling deliveries from different stores but the same segment in the same period can lead to savings on loading costs by decreasing the number of loading gates a single MCV has to approach at the depot. However, savings on loading costs may be exceeded by additional travel costs that arise if deliveries from the same segment but not from the same delivery area are placed on a single tour. To avoid this, the Segment Bundle Operator tries to assimilate patterns of segment-store combinations that are located in the same neighborhood and concern identical segments. It combines the metric for the travel costs between segment-store combinations with the segment similarity. The segment similarity, defined as $\sigma_{si, \tilde{s}j}$, indicates whether the segments considered are equal or not:

$$\sigma_{si, \tilde{s}j} = \begin{cases} 0, & \text{if } s = \tilde{s} \\ 1, & \text{otherwise} \end{cases} \quad (1.19)$$

The relatedness measure $R_{si, \tilde{s}j}^2$ for this operator is denoted by Equation (1.20). Again, we use a weight δ for the two components of $R_{si, \tilde{s}j}^2$. To achieve the desired effect, the weights here are to be chosen differently to the Proximity Operator, with a stronger focus on the similarity of segments than on the relative travel costs of the corresponding stores.

$$R_{si, \tilde{s}j}^2 = \delta \cdot \frac{c_{ij}^{\text{travel}}}{\bar{c}^{\text{travel}}} + (1 - \delta) \cdot \sigma_{si, \tilde{s}j} \quad (1.20)$$

Sales Volume Operator The third score-related operator is based on the overall segment-specific demand of a store for the entire planning period. This total demand is denoted by Ψ_{si} , $s \in S, i \in N$. It may be favorable to align deliveries for stores with comparable demand since we consider heterogeneous stores with different store sizes and sales volumes for each segment. The operator offers the option to copy cost-efficient patterns already found for a segment-store combination to another store with a similar demand structure. Unlike the first two operators, it solely aims to reduce pattern-dependent costs and hence supports the diversification of our search algorithm. Again, the more similar the total demands, the lower the calculated $R_{si, \tilde{s}j}^3$ (see Equation 1.21). This represents the absolute difference of the total demand of the combinations (s, i) and (\tilde{s}, j) .

$$R_{si, \tilde{s}j}^3 = |\Psi_{si} - \Psi_{\tilde{s}j}| \quad (1.21)$$

(ii) Pattern-dependent Cost Operator Cost-intensive patterns may be assigned in the course of the ALNS as they may be favorable in terms of transportation costs. Yet depending on the problem instance's cost structure, the pattern-dependent costs may exceed the savings achieved. Moreover, specific delivery period combinations or delivery frequencies can be extraordinarily costly. We therefore introduce an operator directly focusing on pattern-dependent costs that counteracts this

effect. The operator therefore considers the cost of a chosen pattern p , $p \in P$, for a segment-store combination (s, i) and aims to find a new pattern p' , $p' \in P : p' \neq p$, with lower pattern-dependent costs. Algorithm 2 presents the algorithmic structure of this operator. Please note that Algorithm 2 is similar to Algorithm 1 but not identical. The pattern-dependent cost operator directly considers the cost delta between the current, c_{psi}^{pat} , and the minimum pattern-dependent costs, $\underline{c}_{psi}^{pat}$ of a pattern-segment-store combination (p, s, i) , $\Delta c_{psi}^{pat} = c_{psi}^{pat} - \underline{c}_{psi}^{pat}$. DP \underline{p} , leading to the minimum pattern-dependent costs, is determined as follows: $\underline{p} = \arg \min_{p \in P} [c_{psi}^{pat}]$. The Pattern-dependent Cost Operator evaluates whether a pattern that results in lower costs is available. In *Step 2* of Algorithm 2 we do not select the segment-store combination that promises the highest savings, but the combination that lays ζ^α down the descending cost ranking (c.f. score-related operators). If the minimum pattern-dependent costs during an iteration are not yet reached, a new pattern p' is selected randomly among the set of all patterns P , implying lower pattern-dependent costs. Due to its nature, the Pattern-dependent Cost Operator can be seen as a regulatory operator. For the overall problem it is not effective to select the patterns with minimum costs for each segment-store combination as this would usually lead to high transportation costs. Yet the pattern-dependent costs are one main driver of a successful search, and low-priced patterns have to be utilized in different combinations.

Algorithm 2: Pattern-dependent Cost Operator

Input: Solution \mathbb{S} , set of patterns P , number of segment-store combinations to be changed c , pattern-dependent costs of all combinations c_{psi}^{pat} , degree of randomization α , list O of all segment-store combinations (s, i) , $s \in S, i \in N$

Output: new partial solution \mathbb{S}^* with the updated pattern assignments for all segment-store combinations considered (s, i) , $s \in S, i \in N$

```

1 List  $L = \emptyset$ 
2 while  $|L| < c$  do
3   Step 1: sort all segment-store combinations  $(s, i) \in O$  by  $c_{psi}^{pat}$  in descending order
4   Step 2: draw a random number  $\zeta \in [0, 1]$  and select the segment-store combination  $(s, i)$ , that lays  $\zeta^\alpha$ 
           down the ranking
5   if pattern-dependent costs of current pattern  $p$  of segment-store combination  $(s, i)$  do not equal  $\underline{c}_{psi}^{pat}$ , then
6     Step 3: assign a new pattern  $p'$ ,  $p' \in P$ , to segment-store combination  $(s, i)$ , which is selected
           randomly among all patterns  $P$  with lower  $c_{p'si}^{pat}$  than the current pattern of segment-store
           combination  $(s, i)$ 
7   else
8     continue;
9   Step 4: remove  $(s, i)$  from  $O$ , add  $(s, i)$  to  $L$ 
10 forall  $(s, i) \in O$  do
11 keep the assigned pattern  $p$ ,  $p \in P$ 

```

(iii) Move-related operators Two move-related operators are introduced to increase the diversification of the search algorithm as they do not use a specific search criterion but randomly choose orders to be considered for moves. A move, in this context, is the change of delivery periods while maintaining the identical number of deliveries per week. For example, if a store receives a product segment on day 1 and 2, the Move-One Operator moves one and only one of these delivery days to another period where currently a delivery is not intended. So the DP may change to deliveries on day 2 and day 5. The DP featuring a delivery frequency of $|T|$ times within the planning horizon

stays unconsidered since none of the delivery days scheduled can be moved. The pseudo-code of the Move-One Operator is denoted in Algorithm 3.

The Move-Two Operator moves two delivery periods planned to two delivery periods not yet scheduled. The DPs that intend deliveries on $|T|$, $|T| - 1$ or 1 periods of the planning horizon are not considered since no feasible moves exist for these patterns. We use the Move-One Operator in those cases, except when the delivery frequency equals $|T|$.

Algorithm 3: Move-One Operator

Input: Solution \mathbb{S} , set of patterns P , number of segment-store combinations to be changed c , list O of all segment-store combinations (s, i) , $s \in S, i \in N$
Output: new partial solution \mathbb{S}^* with the updated pattern assignments for all segment-store combinations considered (s, i) , $s \in S, i \in N$

- 1 List $L = \emptyset$
- 2 **while** $|L| < c$ **do**
- 3 *Step 1:* randomly select a segment-store combination $(s, i) \in O$
- 4 **if** the frequency of the current pattern p of (s, i) is unequal $|T|$, **then**
- 5 *Step 2:* randomly select one delivery period t_1 , $t_1 \in \{T \mid a_{pt_1} = 1\}$ of the current pattern p of segment-store combination (s, i)
- 6 *Step 3:* move the delivery from period t_1 to period t_2 , $t_2 \in \{T \mid a_{pt_2} = 0\}$ in the current pattern p of (s, i) and assign the resulting pattern p' to (s, i)
- 7 *Step 4:* remove (s, i) from O , add (s, i) to L
- 8 **forall** $(s, i) \in O$ **do**
- 9 keep the assigned pattern p , $p \in P$

(iv) Random Operator The Random Operator is introduced as an additional diversification operator that also changes the patterns. It randomly selects segment-store combinations and randomly assigns a new pattern from all possible patterns.

1.5.2.2 Stage 2: LNS for Solving the Routing in each Period

Stage 2 within the improvement phase of the entire algorithm addresses the routing problem. This stage solves the MCVRP for each period t , $t \in T$, assuming the DPs selected in Stage 1. We apply the LNS framework suggested by Hübner and Ostermeier (2019) for solving the MCVRP since they assume an equivalent cost structure to ours. In addition, the LNS approach enables high-quality solutions in short computation times (see also Derigs et al. (2011)) that are particularly relevant in our case since we need to solve the MCVRP in each iteration for each period of the planning horizon. The LNS approach of Hübner and Ostermeier (2019) uses the Savings Algorithm by Clarke and Wright (1964) to generate an initial solution for the routing in each period. This is conducted equivalently as described in Section 1.5.1. Based on the initial solution, Shaw Removal and regret- k insertion are used within the LNS. The Shaw Removal is based on Shaw (1997), but is modified to consider the joint delivery of multiple segments. In the present implementation of the LNS we further adapt the Shaw Removal operator to account for the dynamic structure of the delivery sizes. The algorithmic structure presented in Algorithm 1 resembles the structure of Shaw Removal as it also serves as the basis for our score-related operators. The associated relatedness

measure is given in Equation (1.22).

$$R_{si,\tilde{s}j}^S = \mu \cdot \frac{c_{ij}^{\text{tran}}}{c^{\text{tran}}} + \nu \cdot \sigma_{si,\tilde{s}j} + \xi \cdot \frac{|o_{psit} - o_{\tilde{p}\tilde{s}jt}|}{o^{\text{max}}} \quad (1.22)$$

$R_{si,\tilde{s}j}^S$ takes three metrics into account: transportation costs, product segment and delivery sizes, weighted by μ , ν , and ξ , respectively. The metrics for transportation costs and segment similarity are identical to those used in the Segment Bundle Operator. Additionally, the delivery size is included as swapping deliveries with similar sizes results in the faster generation of new feasible solutions. Delivery sizes are compared using the difference in size of deliveries o_{psit} and $o_{\tilde{p}\tilde{s}jt}$ (for the given delivery period t and the corresponding patterns p and \tilde{p}) in relation to the highest delivery quantity across all deliveries (o^{max}). After the defined number of deliveries has been removed (see Algorithm 1), removed deliveries are reinserted applying regret- k insertion (Ropke and Pisinger 2006). It calculates the regret values, i.e., differences, between the best insertion possibility of a delivery and the k -best options. The delivery with the highest difference (regret) is inserted in each iteration. This allows a more foresighted insertion that takes future costs into account. The Regret Operator is indispensable for the search as it significantly improves the solution quality of MCVRPs (see Derigs et al. (2011)). Finally, Record-To-Record Travel as introduced by Dueck (1993) is used as an acceptance criterion for the LNS. Accordingly, a new solution is accepted as a new incumbent solution if it lays within a defined deviation (D) from the best solution found so far. The LNS terminates after a predefined number of iterations without a solution improvement.

1.5.3 Evaluation and Control Mechanism of entire Algorithm

Simulated annealing While testing our approach we found that the search process tends to get trapped in local minima. We therefore use a Simulated Annealing framework to govern the search and enable broader diversification. Accordingly, a new solution S^* found within the improvement heuristic is accepted if it is better than the best-known solution so far, S_{best} , or the incumbent solution, S_{inc} . Further, for a higher degree of diversification, a worse solution is accepted as an incumbent solution with the probability $e^{-\frac{f(S') - f(S_{\text{inc}})}{E}}$. This probability is then decreased in the course of the search process. The temperature $E > 0$ is initialized using E_{start} and decreased in each successive iteration by the cooling rate factor $d \in]0; 1[$. For the calculation of E_{start} we adapted the method of Ropke and Pisinger (2006) to fit the requirements of the PMCVRP. Consequently, E_{start} is set such that a solution subsequently obtained is accepted with a probability of 0.5, i.e., $E_{\text{start}} = -\frac{g \cdot f(S_{\text{start}})}{\ln 0.5}$, if its objective function value is g percent worse than the starting solution.

Additional diversification Apart from being used as operator within the ALNS, the Random Operator is deployed as an additional tool of diversification. This is why we introduce a reset border λ . If λ iterations are made without a new best solution being found, the Random Operator is used, changing a high number of segment-store combinations and thereby destroying a large part of the current solution.

Termination criterion If the number of ALNS iterations without a new best solution found reaches a predefined limit, the search process is stopped. This limit is independent of the reset border λ and is never readjusted in the course of the solution approach.

Adaptive operator selection The final step of each ALNS iteration is the adaptation of operator weights used within the ALNS approach. As stated above (Section 1.5.2.1), the operator selection within the ALNS is based on individual weights for each operator. As proposed in Ropke and Pisinger (2006), we use a roulette wheel selection principle where the probability Φ_h of operator h , $h \in \{1, 2, \dots, 7\}$ (see Table 1.2) being selected for the current iteration is determined by its weight ρ_h and $\Phi_h = \frac{\rho_h}{\sum_h \rho_h}$. In the beginning of the search, the likelihood of selection is equal for all operators, i.e., the weights of all operators are set to 1. Later, the weights are adjusted depending on their performance in the previous search leg. Note, a single *search leg* is defined by a specific number of consecutive iterations. A score Θ_h is introduced to measure the performance of operator h during the last search leg. The performance of the operators is measured by evaluating the overall solution obtained, i.e., the total costs according to Equation (1.1). Θ_h is increased by θ_1 if the operator results in a new best solution, by θ_2 if the operator results in a new incumbent solution, and by θ_3 if the operator results in a new solution, but is not accepted. At the end of each search leg, the weights ρ_h for every operator h are updated according the average scores achieved, $\frac{\Theta_h}{\Omega_h}$ using Equation (1.23). Ω_h denotes the number of times operator h is selected in the last search leg. The magnitude of change for the weights is controlled by the *smoothing factor* $\tau \in]0; 1[$. After all new weights have been calculated, all Θ_h are reset to zero for the next search leg.

$$\rho_h = (1 - \tau) \cdot \rho_h + \tau \cdot \frac{\Theta_h}{\Omega_h} \quad \forall h \in \{1, \dots, 7\} \quad (1.23)$$

Post-optimization of routing After the stop criteria of the ALNS are met, we apply a post-optimization step to improve the final routing solutions, i.e., the MCVRPs for each period of $|T|$. We therefore apply an extended LNS to the routing problem of each period, increasing the LNS search limit of unsuccessful iterations significantly. Please note that using a higher limit for the LNS is only feasible at the end of the ALNS search, as the LNS is frequently applied during the search (i.e., $|T|$ times per ALNS iteration) and runtimes would increase exponentially.

1.6 Numerical Experiments

Numerical experiments are applied to evaluate the performance of our solution approach and the interdependence between the planning of DPs and MCVs. The runtime is analyzed in Section 1.6.1. Section 1.6.2 compares our approach to the results of Holzapfel et al. (2016) to provide a benchmark with regard to solution quality. The impact of using MCVs instead of SCVs for the determination of DPs and the overall solution structure is assessed in Section 1.6.3. Finally, we apply our approach to a real-world case at a major German retailer in Section 1.6.4. Table 1.3 gives an overview of the numerical experiments and the data sets used.

| Section | Experiments and purpose | Stores | Segments | Data set | Number of instances |
|---------|---|-------------|----------|--|---------------------|
| 1.6.1 | Runtime analysis | 25, 50, 100 | 3 | Solomon (1987) | 18 |
| 1.6.2 | Single segment benchmark | 30, 40 | 1 | Holzapfel et al. (2016) | 120 |
| 1.6.3 | Value of MCV integration and joint deliveries for DP planning | 50 | 2-4 | Simulated data, informed by case study | 20 |
| 1.6.4 | Case study | 376 | 3 | Real-world data | 1 |

Table 1.3: Overview of numerical experiments

Data applied Each test instance is defined by the number of stores (and their spatial distribution), number of segments (and the number of their products) as well as the planning horizon, which in turn determines the number of possible DPs. If not stated otherwise, we consider five days ($|T| = 5$) as planning horizon with all possible DPs (i.e., $2^5 - 1 = 31$ combinations). The number of possible DPs may, however, be reduced due to non-feasible combinations, i.e., we check if a DP violates vehicle or store capacities (Q^{ved} and Q_i^{recmax}) for all given segment-store combinations. We apply a daily demand for each segment-store combination and specify the ranges for each data set. The daily demand for the specific weekdays depends on the weekly seasonality obtained from data of a benchmark case (see Section 1.6.2) and a real case study (see Section 1.6.4). The shelf capacity for each product was set equally for all stores in all tests. It was determined using the ratio of average weekly product demand to product shelf capacity as given in Holzapfel et al. (2016). Vehicle capacity is set to 2,700 transportation units (TU). Also, we adopt the empirical cost parameter setting by Holzapfel et al. (2016) for store- and DC-related costs as well as MCV-related loading and unloading cost parameters from Hübner and Ostermeier (2019). The exact values of case study related data are subject to non-disclosure agreements.

Implementation details The algorithm-specific parameter setting used for our experiments is summarized in Table 1.4. We adopted the corresponding values reported in literature (see Table 1.4) for the majority of parameters as these yield excellent results for our setting. The weights δ and ϵ in the bundle operator segment have been tuned within our tests. The cooling rate (d) was adjusted compared to values reported in Ropke and Pisinger (2006) due to a differing number of iterations and different objective value ratios. The number of segment-store combinations (c) was chosen depending on the corresponding problem size as depicted in Table 1.5. All experiments were executed with a limit of 3,000 iterations without a new best solution found. The adaptive weights were adjusted after every search segment of 50 iterations. The LNS for the evaluation of the MCVRPs terminates after 100 unsuccessful iterations. The extended LNS, which is applied only in the post-optimization when the ALNS is terminated, stops after 2,000 unsuccessful iterations. Our algorithm is also built on a stochastic search procedure. To balance this out, we apply the same instance multiple times, depending on the data set used. The frequency is denoted in the respective tests. The algorithm described in Section 1.5 was implemented in Java 8 and used for all following experiments.

| Parameter | Value | Function | Origin |
|--------------------------------|---------------|--|------------------------------|
| α | 6 | Degree of randomization | Ropke and Pisinger (2006) |
| β, γ | 0.8, 0.2 | Weights of proximity operator at ALNS | Derigs et al. (2011) |
| δ, ϵ | 0.3, 0.7 | Weights of segment bundle operator at ALNS | Own experiments |
| μ, ν, ξ | 0.6, 0.2, 0.2 | Weights at LNS | Hübner and Ostermeier (2019) |
| k | 2 | Regret insertion parameter at LNS | Hübner and Ostermeier (2019) |
| D | 0.003 | Deviation allowed at LNS | Hübner and Ostermeier (2019) |
| d | 0.99975 | Simulated annealing cooling rate | Own experiments |
| g | 0.03 | Start temperature control parameter | Ropke and Pisinger (2006) |
| λ | 200 | Reset border | Derigs et al. (2011) |
| $\theta_1, \theta_2, \theta_3$ | 33, 9, 11 | Operator score increase for new solutions | Ropke and Pisinger (2006) |
| r | 0.1 | Reaction factor for operator weight adjustment | Ropke and Pisinger (2006) |

Table 1.4: Algorithm parameters used

| Instance size (# stores) | No. of segment-store combinations | |
|-----------------------------|-----------------------------------|---------|
| | Minimum | Maximum |
| 25 | 5 | 15 |
| 50 | 5 | 20 |
| 100 | 5 | 30 |
| >200 | 5 | 50 |

Table 1.5: Setting of parameter c

1.6.1 Runtime Analysis

Runtime data We use the VRPTW data sets provided by Solomon (1987) to analyze the computation times of our solution approach. The data set comprises instances with 25, 50 and 100 customers and is subdivided into three categories: (C) clustered stores, (R) uniformly random distributed stores and a (RC) mixture of both. Solomon (1987) provides two different spatial distributions of stores for each category, resulting in six instances for each of the classes with 25, 50 and 100 stores. We use the number and the spatial distribution of customers from these data sets. The actual distances are multiplied such that the delivery area resembles a realistic distribution area in retail practice, similar to our case study (see Section 1.6.4). This allows maintenance of the general cost parameter setting as otherwise the share of travel costs would be underestimated. We set up three product segments for a five-day week.

Runtime comparison The runtime analysis is summarized by each instance class in Table 1.6. As the definition of DPs is a tactical planning problem, the runtimes are still within an acceptable range, considering DPs are not defined on a weekly but monthly or yearly basis. The runtime strongly increases with the increase of problem sizes. This is due to the increase in segment-store combinations and the resulting complexity for pattern and routing decisions. More than 90% of computational time is consumed by the regret- k -insertion heuristic within the LNS for the daily routing. However, the regret- k -insertion is substantial to obtain good routing solutions as it significantly improves the solution quality of the search (see also Derigs et al. (2011)). The regret value has to be recalculated for each insertion and each possible position on trucks and therefore consumes significant computation time. Several parameters impact the runtime of the regret- k -

insertion heuristic. The number of orders to be removed, the degree of regret (k), and the overall number of orders on each day are the most important drivers. In our problem, the number of orders per day changes dynamically and also the corresponding delivery volume. In contrast to other PVRPs, both delivery frequency and days are permanently changed within our approach.

| No. of stores | 25 | | | 50 | | | 100 | | |
|------------------------------------|------|------|------|------|------|------|-------|-------|------|
| No. of segment-store combinations | 75 | | | 150 | | | 300 | | |
| Type of spatial store distribution | C | R | RC | C | R | RC | C | R | RC |
| Runtime, <i>min</i> | 0.13 | 0.21 | 0.16 | 0.35 | 0.57 | 0.34 | 1.83 | 1.30 | 1.09 |
| Runtime, <i>average</i> | 0.36 | 0.65 | 0.57 | 1.50 | 1.60 | 0.93 | 4.77 | 4.72 | 3.93 |
| Runtime, <i>max</i> | 0.74 | 1.07 | 0.84 | 3.33 | 2.80 | 1.94 | 15.48 | 10.17 | 9.28 |

Table 1.6: Total computation times for different problem sizes, in hours

1.6.2 Comparison with Single Segment Benchmark

Benchmark approach The effectiveness of our approach is shown by a comparison with Holzapfel et al. (2016). This approach is a special variant of our problem (multi-period, single segment and SCVs), and the only available benchmark for our setting. The authors solve the allocation of DPs by an optimally solved general assignment problem (GAP) while approximating the resulting transportation costs using the approach of Fisher and Jaikumar (1981). Contrary to our approach, day-to-day vehicle routing is not part of their solution procedure. They assume stable base tours, i.e., assignments of store orders to vehicles for each day of the planning horizon. In order to make a fair comparison, we re-evaluated the DP assignment from Holzapfel et al. (2016) using our general approach to solve the VRPs on each day of the planning horizon. This means we also apply the LNS approach proposed in the present paper for each instance derived from Holzapfel et al. (2016) to further improve the results presented there. In doing so, we apply the extended LNS for each delivery day five times and keep the best daily solution found. This procedure entirely corresponds to the assumptions and the implementation of our overall modeling and solution approach.

Benchmark data Holzapfel et al. (2016) apply scenarios for a single segment across six delivery days with 10, 20, 30 and 40 stores and three different delivery area sizes. The stores are randomly located within a delivery area of 50 km \times 50 km (“Metropolitan”), 200 km \times 200 km (“District”) or 400 km \times 400 km (“State”). All demand, store and cost parameters in Holzapfel et al. (2016) are set according to empirical data of a partner company as well as to data collected by Kuhn and Sternbeck (2013) and Sternbeck and Kuhn (2014). In the benchmark data, not all of the $2^6 - 1 = 63$ DPs are feasible for all stores due to vehicle (Q^{vec}) or receiving capacities at stores (Q_i^{recmax}). Consequently, around one-third of DPs can be excluded upfront with respect to individual segment-store combinations. Since smaller problem classes are not relevant for our application, we focus on instances with 30 and 40 stores, totaling 120 instances, for the benchmark calculations.

Benchmark comparison The overall approach suggested in the present paper achieved costs savings in all instances but one (see Figure 1.5) compared to the adapted Holzapfel et al. (2016)

approach.

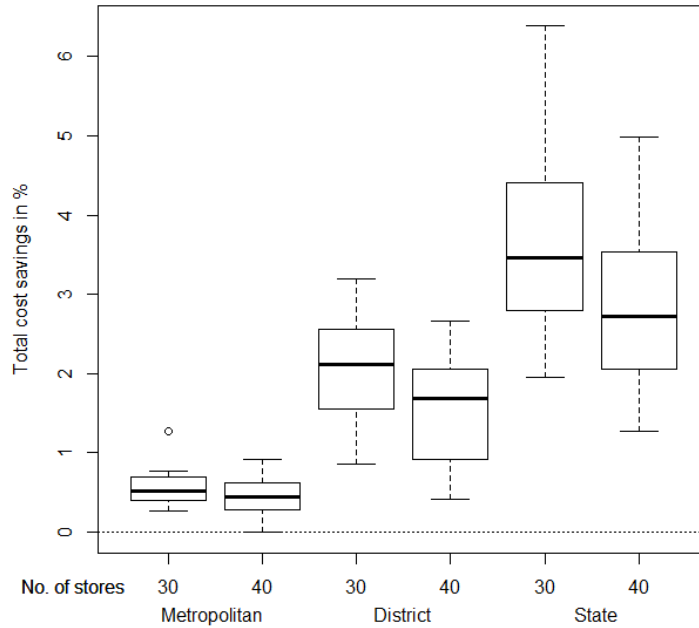


Figure 1.5: Distribution of total cost savings depending on delivery area and number of stores in comparison to the adapted benchmark of Holzapfel et al. (2016) (best of 5 runs)

In detail, two effects can be observed. First, the larger the delivery area, the greater are the improvements achieved. With increasing delivery area size and thus higher travel costs, routing becomes more important. Since Holzapfel et al. (2016) approximate the travel costs, actually solving the VRP gains importance. Secondly, it can be observed that the average improvement for instances with 40 stores is slightly lower than with 30 stores. This can be attributed to the increasing impact of DP selection. As more customers are involved, it is more important to find the optimal DP assignment in order to exploit the bundling effects in transportation. As Holzapfel et al. (2016) solve the allocation of DPs optimally using approximations for the resulting travel costs, it becomes more difficult for our approach to generate additional improvements via delivery pattern assignment. Table 1.7 shows the cost savings in percent of total costs obtained by Holzapfel et al. (2016). Our results show that total cost savings originate almost entirely from travel costs, whereas the relative pattern-dependent cost savings are on average only around 0.13%.

| Delivery area | Metropolitan | | District | | State | | Average |
|-------------------------------|--------------|--------|----------|--------|--------|--------|---------|
| Number of stores | 30 | 40 | 30 | 40 | 30 | 40 | |
| Total cost savings | 0.56% | 0.44% | 2.05% | 1.55% | 3.71% | 2.81% | 1.85% |
| Savings share of ¹ | | | | | | | |
| Travel cost | 0.50% | 0.36% | 1.93% | 1.46% | 3.58% | 2.73% | 1.76% |
| Unloading cost | 0.03% | -0.06% | -0.06% | -0.05% | -0.03% | -0.06% | -0.04% |
| Pattern-dependent cost | 0.03% | 0.13% | 0.18% | 0.14% | 0.16% | 0.13% | 0.13% |

¹ Relative share of savings in relation to total costs of Holzapfel et al. (2016)

Table 1.7: Cost savings in comparison to the adapted benchmark of Holzapfel et al. (2016) (best of 5 runs)

Concerning computation times, Holzapfel et al. (2016) report an average of 2.7 minutes in addition to a pre-processing time of about 5 minutes for instances with 40 customers, while the computation times of our approach amount to an average of 5.23 minutes per single run for the same instances.

In conclusion, we observe that our solution approach is able to solve the related benchmark problem effectively. It improves the benchmark solutions by around 1.85% on average across all problem classes when applying the best solution of five runs. The average of all five runs still improves the results by around 1.25% across all problem classes. Moreover, we would like to note that our approach aims at problems with multiple segments and the corresponding cost savings when segments are jointly delivered. This effect is not taken into account within the benchmark comparison as only a single segment is considered.

1.6.3 Planning of DPs across Segments with MCVs

A core aspect of our work is the use of MCVs for distribution and the corresponding impact on determining DPs. The different product flows and consequently joint deliveries are only possible with MCVs. In line with this, we compare the results of our solution approach with MCVs to a solution using SCVs only where the product segments are distributed separately. In the SCV scenario we apply our solution approach for each product segment separately, thus generating independent single segment solutions. The scenario with joint planning and delivery is denoted as “MCV” and the separate one as “SCV”.

Test data For the following analysis we use simulated data sets. It comprises four scenarios with five instances for each, totaling 20 data sets. The generation of these data sets is informed by the real-world data from our case study. All instances comprise 50 stores, which are randomly located around a DC within a delivery area of 230 km \times 180 km. Demands are simulated for three product segments. The total demand across the delivery week follows a normal distribution with $\mu = 1,400$ and $\sigma = 800$. Stores generally feature – depending on their size – different sales volumes across segments. The weekly demand is therefore randomly multiplied by a factor of 0.5 (low), 1.0 (medium) or 1.5 (high) to simulate different store sizes. The daily demand is subject to weekly seasonality with the distribution factors $\{0.149, 0.233, 0.205, 0.211, 0.202\}$ for a five-day week. The shelf capacity for each product is set equally for all stores, but randomly across products, according to the ratios of demand to shelf capacity reported by Holzapfel et al. (2016). Based on the medium store size, this results in normally distributed shelf capacity with $\mu = 1,000$ and $\sigma = 400$. As the available real-world data lacks information on picking and store receiving capacities, these are set as unconstrained. We apply our heuristic five times to each instance and compare the best results achieved.

The segment share of the total order volume may have a major impact. One could expect that the more segments are available, the higher the potential for cost savings through the joint delivery of segments. We therefore apply four different scenarios. The demand share of each segment for each scenario is given in Table 1.8. In Scenarios 1 to 3, each store follows the segment shares indicated. In Scenario 1 for example, Segments 1, 2 and 3 comprise 80%, 10% and 10% of the total

1 Optimizing Routing and Delivery Patterns with MCVs

order volume, respectively. Scenario 4 combines Scenarios 1 to 3 by randomly assigning one of the given scenarios to each store.

| | Scenario 1 | Scenario 2 | Scenario 3 | Scenario 4 |
|-----------|------------|------------|------------|------------|
| Segment 1 | 80% | 50% | 33.3% | 54.4% |
| Segment 2 | 10% | 30% | 33.3% | 24.4% |
| Segment 3 | 10% | 20% | 33.3% | 21.1% |

Table 1.8: (Mean) segment share of total order volume

Results We first analyze the overall cost structure for all scenarios. Table 1.9 illustrates the share of cost components of the total costs across the different scenarios.

| Scenario | 1 | | 2 | | 3 | | 4 | |
|-------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | MCV | SCV | MCV | SCV | MCV | SCV | MCV | SCV |
| Loading costs | 5.6% | 1.7% | 5.7% | 1.7% | 5.7% | 1.6% | 5.3% | 1.6% |
| Travel costs | 26.5% | 28.9% | 26.7% | 30.9% | 27.4% | 34.3% | 27.7% | 30.2% |
| Unloading costs | 4.3% | 5.5% | 4.2% | 6.3% | 4.1% | 6.5% | 4.6% | 5.7% |
| Pattern-dependent costs | 63.6% | 63.8% | 63.4% | 61.0% | 62.8% | 57.6% | 62.2% | 62.4% |

Table 1.9: Share of costs components of total costs

The cost structure of the MCV and SCV scenarios identify pattern-dependent costs as main cost driver of the PMCVRP. Pattern-dependent costs account for almost two-thirds of total costs with an average of 63% (MCV) and 61% (SCV). The share of travel costs on the other hand is only half as high with 27% (MCV) and 31% (SCV). This underlines the importance of DP selection for a low cost solution. We further compare potential cost savings and other performance indicators across all scenarios when all segments are jointly planned and MCVs instead of SCVs are used. Tables 1.10 and 1.11 summarize our findings. In all of our scenarios, the joint planning and usage of MCVs results in significant cost reductions, with up to 15% of total cost savings (Scenario 3).

| Scenario | 1 | 2 | 3 | 4 |
|-------------------------------|--------|--------|--------|--------|
| Total costs saving | 8.28% | 13.12% | 15.15% | 10.60% |
| Savings share of ¹ | | | | |
| Loading costs | -3.43% | -3.30% | -3.20% | -3.12% |
| Travel costs | 4.67% | 7.73% | 11.14% | 5.44% |
| Unloading costs | 1.60% | 2.69% | 2.98% | 1.58% |
| Pattern-dependent costs | 5.44% | 6.00% | 4.23% | 6.69% |

¹ Relative share of savings in relation to total SCV costs

Table 1.10: Cost savings of MCV compared to SCV

As expected, loading costs increase in all scenarios when MCVs instead of SCVs are used. They occur for each segment loaded on a vehicle. The loading costs for MCV vs. SCV are approximately two times higher. This shows that on nearly all tours three segments are combined when MCVs are available. The main savings are achieved by lower pattern and travel costs. They originate in the joint delivery of segments and are therefore dependent on the demand scenario considered.

Pattern-dependent savings are highest when the segment demand structure is most heterogeneous (Scenarios 2 and 4). DPs across the different segments are aligned and thus costs decrease. Travel cost savings are higher in scenarios where the segment volumes are equally spread across segments as shown in Scenario 3. With equally spread volumes, it is very likely that different segments are supplied together, whereas in Scenario 1 (80%/10%/10% volume shares), bundling effects across segments are limited. The same can be observed with respect to unloading costs: most stops at stores can be avoided in Scenario 3 (306 (SCV) vs. 164 (MCV) stops).

| Scenario | 1 | | 2 | | 3 | | 4 | |
|------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | MCV | SCV | MCV | SCV | MCV | SCV | MCV | SCV |
| ∅ number of orders delivered | 412.8 | 251.2 | 463.2 | 313.0 | 411.8 | 306.0 | 389.4 | 274.2 |
| ∅ number of tours | 29.0 | 29.0 | 30.2 | 30.6 | 27.8 | 28.0 | 28.2 | 29.0 |
| ∅ number of orders/tour | 14.2 | 8.7 | 15.3 | 10.2 | 14.8 | 10.9 | 13.8 | 9.5 |
| ∅ number of stops | 178.4 | 251.2 | 180.2 | 313.0 | 164.8 | 306.0 | 198.2 | 274.4 |
| ∅ capacity utilization | 89.9% | 88.5% | 89.3% | 88.3% | 88.9% | 88.7% | 90.5% | 87.9% |
| ∅ delivery frequency | 2.75 | 1.67 | 3.09 | 2.09 | 2.74 | 2.04 | 2.60 | 1.83 |

Table 1.11: Solution structure of MCV compared to SCV, entire planning period

The corresponding solution structure for each scenario is further depicted in Table 1.11. The total number of deliveries differs as we determine the delivery frequency and therefore the split of total demand across weekdays. The most striking impact of MCVs on the solution structure is increasing delivery frequency and therefore more deliveries in total. For instance, the average number of deliveries rises in Scenario 1 from around 251 to 412 deliveries, an increase of over 64%. The average number of tours remains relatively stable in all scenarios, and consequently the number of orders delivered per tour increases. Interestingly, a greater increase in delivery frequency does not lead to a greater decrease in pattern-dependent costs. Considering Scenario 1, the frequency increases from 1.67 to 2.75 and pattern-dependent costs are reduced by 5.44%. In contrast, Scenario 4 reveals a reduction of 6.69%, while the increase in delivery frequency is lower when MCVs are used (1.83 to 2.60 deliveries). This can be attributed to the fact that the store-individual optimal delivery frequencies are not generally chosen when using SCVs or MCVs. Instead, a different DP and frequency is chosen as another option enables a higher total cost saving (e.g., due to reduced transportation costs). Also, the average capacity utilization per tour improves by about 1.27% on average when using MCVs, as there are more deliveries, and thus more possible loading combinations.

1.6.4 Case Study

Case study data To conclude our numerical analysis, we present a case study with a major German retailer. Our partner company uses MCVs for distribution but had not considered the impact of MCVs on the determination of DPs. We therefore apply our approach to the given problem data to compare the selection of DPs when the benefits of MCVs are taken into account during the decision process. The case study covers a representative five-day week with orders of 376 stores to be delivered from a single depot. Most stores have a relatively small demand such that most of them are only served once a week. The order structure is heterogeneous, with about

50% of stores ordering one segment, about 20% ordering two segments, and about 30% ordering all three segments. Moreover, the most frequently ordered segment accounts for around 80% of the total order volume. The other two segments account for roughly half of the remaining volume each. Consequently, this order structure resembles Scenario 1 above. The order volumes of the three segments follow a weekly seasonality. Regarding store sizes, the 10% of stores with the highest sales also account for more than 40% of the order volume. This means that the majority of stores have a small demand volume. As in the previous data sets, the shelf space data was supplemented based on an average-sized store demand assuming the same ratio between product demand and shelf capacity as in Holzapfel et al. (2016).

In the following we analyze the potential cost savings if the retailer coordinates DP and MCV planning. In doing so, we compare the retailer’s approach denoted as “status quo approach” with our approach denoted as “integrated approach”. The status quo approach equals our approach, but assumes the DPs currently applied by the retailer. We therefore assume that the retailer already applies the same MCVRP solution approach as we do. This means we only evaluate the effect of planning DPs across segments and do not mix this with potential effects resulting from different approaches used to solve the MCVRP. We apply both approaches ten times and compare the respective best solution.

Results Table 1.12 presents the resulting cost savings in % of the status quo total costs; and Table 1.13 displays some performance indicators and reveals insights into the respective solution structure achieved. The runtime of our integrated solution approach amounts to an average of 2.89 hours.

| | |
|-------------------------------|-------|
| Total cost savings | 7.68% |
| Savings share of ¹ | |
| Loading costs | 0.36% |
| Travel costs | 4.08% |
| Unloading costs | 0.97% |
| Pattern-dependent costs | 2.27% |

¹ Relative share of savings in relation to total costs of status quo

Table 1.12: Cost savings of the integrated approach compared to status quo

| | Status quo approach | Integrated approach |
|-------------------------------------|---------------------|---------------------|
| Number of orders delivered per week | 709 | 630 |
| Number of tours per week | 24 | 22 |
| Number of stops per week | 418 | 381 |
| ∅ capacity utilization | 86.7% | 94.6% |

Table 1.13: Solution structure resulting from the integrated approach compared to the status quo approach

The integrated approach that jointly determines DPs across segments results in total cost savings of 7.68% compared to the status quo approach. The savings mainly result from improved

pattern and travel costs, which are facilitated by improved DP assignment. Only the segment-store combinations with very high demand are supplied twice a week. Almost all other combinations are delivered once a week. The number of stops and tours required are therefore reduced, resulting in higher vehicle capacity utilization. The reduction of delivery tours is an essential driver of cost reduction as it leads to personnel cost savings. This means that at our case company the number of delivery tours should be further decreased as the savings outweigh the increase in in-store logistics costs although the company's solution already reveals a low average delivery frequency. In conclusion, we can state that an integrated solution approach enables better evaluation of the complete planning problem. When delivery options with MCVs are taken into account, DPs can be adjusted to align deliveries across the complete planning horizon and to ultimately reduce total costs. This can result in both increasing (see Section 1.6.3) or decreasing frequency, as shown in the case study.

1.7 Conclusion

In this paper we introduced a new MCVRP variant, the PMCVRP, that addresses the selection of delivery patterns when MCVs are used for distribution. The PMCVRP is a multi-period MCVRP applied for grocery distribution. However, the practical relevance is not limited to this application as it can easily be adapted to other application areas in which periodicity of deliveries is relevant (e.g., fuel distribution or agricultural problems). The problem presented combines the research on DP planning with an MCVRP and consequently closes an existing gap in literature by identifying new options for delivery planning. More precisely, the objective of our work is to highlight the impact on DP planning when the deliveries of different product segments can be combined across the planning horizon when using MCVs. This paper identifies the decision-relevant processes and corresponding costs for both the choice of patterns and the use of MCVs, and presents a formal model description. The resulting problem is solved using an ALNS approach for assigning patterns and an LNS for solving the routing. It is tailored to the given problem specifics. The performance of the algorithm proposed is compared to an existing approach in literature to show its efficiency and effectiveness. This revealed that our approach is able to improve given solutions for DP planning. In subsequent numerical experiments we analyze the interdependencies between routing with MCVs and DP planning. We show that, depending on the given problem characteristics, the PMCVRP leads to a different solution structure (i.e., altered delivery frequencies) and reduces total costs compared to the prevailing planning with SCVs as it combines different product flows and adjusts the corresponding patterns of stores accordingly. Finally, we consider a case study with a major German retailer for the supply of stores with small order volumes. The case study shows the practical relevance of our approach and improves the planning solution of the retailer by around 8% if DP planning is solved using MCVs.

The research on MCVRPs has steadily grown over the past years and our work further contributes to this field by closing another existing gap in literature. However, there are still numerous possibilities for future research. First, we consider an MCVRP for master route planning and assume given demands for an average week. In this context, the consideration of stochastic demand

could further improve the planning as a more realistic evaluation of costs would be possible considering realistic demand fluctuations. In general, the consideration of stochastic demands is still neglected in most MCVRP applications (see Ostermeier et al. (2021)). Second, our cooperation with industry shows that due to given economic developments and changed conditions, the existing delivery fleet usually consists of heterogeneous vehicles for different purposes, including both MCVs and SCVs. Consequently, the consideration of a heterogeneous fleet within the PMCVRP would be a valuable next step. Third, the PMCVRP aims at minimizing total costs, consisting of pattern- and routing-dependent cost factors. The consideration of further impacts on profits (e.g., service level agreements, tardy deliveries) as well as ecological aspects (energy consumption, joint vs. split delivery) would be a valuable avenue for future research directions. Lastly, the ALNS and LNS approaches perform well when solving the PMCVRP and MCVRP, respectively. Recently, other solution approaches are suggested that show promising results in related application areas, such as population-based search algorithms relating to waste collection (see Rabbani et al. (2016)).

Contribution 2 - The Vehicle Routing Problem with Availability Profiles

2 The Vehicle Routing Problem with Availability Profiles

Stefan Voigt, Markus Frank, Pirmin Fontaine, Heinrich Kuhn

Abstract In business-to-consumer (B2C) parcel delivery the presence of the customer at the time of delivery is implicitly required in many cases. If the customer is not at home, the delivery fails – causing additional costs and effort for the parcel service provider as well as inconvenience for the customer. Parcel service providers typically report high failed-delivery rates, as they have limited possibilities to arrange a delivery time with the recipient. We address the failed-delivery problem in B2C parcel delivery by considering customer-individual availability profiles (APs) that consist of a set of time windows, each associated with a probability that the delivery is successful if conducted in the respective time window. To assess the benefit of APs for delivery tour planning, we formulate the vehicle routing problem with availability profiles (VRPAP) as a mixed integer program (MIP), including the tradeoff between transportation and failed-delivery costs. We provide analytical insights concerning the model’s cost savings potential by determining lower and upper bounds. In order to solve larger instances we develop a novel hybrid adaptive large neighborhood search (HALNS). The HALNS is highly adaptable and also able to solve related time-constrained vehicle routing problems, i.e., vehicle routing problems with hard, multiple and soft time windows. We show its performance on these related benchmark instances and find a total of 20 new best-known solutions. We additionally conduct various experiments on self-generated VRPAP instances to generate managerial insights. In a case study using real-world data, despite little information on the APs, we were able to reduce failed deliveries by approximately 12% and overall costs by 5%.

Minor Revision: Transportation Science

URL (Working Paper): <https://doi.org/10.2139/ssrn.3793033>

2.1 Introduction & Motivation

In the course of digitization, e-commerce revenues have risen steadily worldwide and are expected to increase even further. They are predicted to account for 19.6 % of global retail sales in the business-to-consumer (B2C) segment in 2021, with an annual growth rate of 25.7% (von Abrams 2021). This trend is the main reason why the number of worldwide parcel deliveries exceeded 131 billion in 2020 (Pitney Bowes 2021). The question regarding how customers receive all their online purchases physically, widely known as the *last mile* problem, becomes an even greater challenge as a result. The dominant delivery mode is home delivery due to its convenience for the customer (Hübner et al. 2016). However, the presence of the customer at the time of delivery is often implicitly required (e.g., for security reasons), resulting in the so-called attended home delivery problem (e.g., Agatz et al. 2011). If the customer is not at home, the delivery fails - causing additional costs and effort for the courier, express and parcel service provider (CEP) as well as inconvenience for the customer. Nowadays, several online stores allow customers a choice between alternative CEPs. This means they have a high incentive to avoid failed deliveries since otherwise the customer might change to a competing CEP. Reported failed-delivery rates in this sector (excluding CEPs that practice unsecured deliveries) can surmount 50% (Okholm et al. 2013). As a possible solution, CEPs could use customer-related data to increase the probability that a customer is at home at the time of the delivery. Pan et al. (2017) already showed that such data can be used to minimize failed deliveries in delivery tour planning, but use a sequential optimization approach, i.e., they first minimize failed-delivery costs and then solve the resulting vehicle routing problem with time windows (VRPTW). This approach neglects the potential increase in distance induced by following time windows (TWs), as well as the potentially increasing number of vehicles. Besides this work, the problem of unsuccessful home deliveries, let alone using customer data to reduce them, has received little attention in research so far. To address this routing problem in the context of B2C parcel delivery, we introduce the vehicle routing problem with availability profiles (VRPAP). We consider customer-individual availability profiles (APs) that consist of a set of discrete, non-overlapping TWs. Each TW is associated with a certain probability of the delivery being successful if conducted in the respective TW. We minimize the sum of routing costs and customer-individual expected failed-delivery costs. The VRPAP explicitly addresses the tradeoff between transportation and failed-delivery costs in this way. The VRPAP is a generalization of the VRPTW and therefore \mathcal{NP} -hard. We develop the hybrid adaptive large neighborhood search (HALNS) for solving this problem efficiently, and compare it to related benchmark algorithms from the literature. Further, we conduct several analyses regarding the impact of failed-delivery costs on vehicle routing with simulated and empirical data.

The contributions of our work are as follows: (1) We show how customer-related data can be used to decrease the share of failed deliveries while considering the cost of routing. We explicitly model the tradeoff between costs for routing and failed deliveries, and define a novel problem class in the setting of parcel delivery, i.e., the VRPAP. (2) We model the problem as a mixed integer program (MIP) and (3) analytically provide insights into its theoretical cost savings potential. (4) We develop a highly adaptable metaheuristic solution framework that is suitable not only for

solving practical-size instances of the VRPAP, but also for solving the VRPTW, the VRP with multiple time windows (VRPMTW), and variants of the VRP with soft time windows (VRPSTW). It produces 12 new best-known solutions for VRPMTW and 8 for VRPSTW benchmark instances. (5) Experiments show the cost savings potential when CEPs consider APs of their customers during the delivery process.

The remainder of this paper is structured as follows: Section 2.2 details the problem setting considered. Section 2.3 gives an overview of research areas concerning relevant application- and model-specific literature. Section 2.4 then presents the VRPAP model formulation. Section 2.5 outlines the HALNS, and Section 2.6 presents the numerical analyses. Section 2.7 finally summarizes our work and indicates further research directions.

2.2 Problem Description

In the following we introduce the VRPAP. Section 2.2.1 describes the delivery process within this context. Section 2.2.2 then details the failed-delivery issue and describes how APs can contribute to increasing the rate of successful deliveries. Afterwards, Section 2.2.3 derives the problem-specific and decision-relevant costs. Lastly, Section 2.2.4 summarizes the operational planning problem considered.

2.2.1 Delivery Process

In e-commerce B2C sales, customers can choose between home delivery and a large set of alternative collection points such as parcel shops, parcel stations, or parcel lockers. Nevertheless, the dominant delivery mode remains home delivery. In this case, the CEP builds round trips from a single depot to a number of customers' home addresses. The tours are carried out by drivers in trucks that bring the deliveries right up to the customer's front door. At each customer, the driver needs to get out of the truck, pick out the customer's parcel, go to the customer's door, wait for the customer to open, hand over the parcel and return. The service time required to perform these tasks depends on the local conditions and varies significantly. In the majority of cases, customers have to receive their delivery in person as the customer's signature is required as a confirmation of receipt (e.g., Hermes 2020). Even for CEPs that in general practice unsecured or unattended home deliveries, the presence of the customer is mandatory for certain deliveries, e.g., for sensitive or highly valuable goods. Unattended home delivery requires a secure place to deposit the parcel such as an accessible storage location. Although this place may theoretically be available, the CEP may not have the customer's consent to deposit their parcels there. So in most cases the delivery fails if the customer is not at home at the time of delivery.

2.2.2 Failed Deliveries & Availability Profiles

Generally, the *failed-delivery rate* is not a standardized performance metric in the CEP industry. CEPs may include all possible reasons for the delivery attempt to fail in this measure, apart from the customer being unavailable, e.g., the delivery address being wrong, not locatable or inaccessible,

or the customer refusing to accept the delivery. At the other extreme, some CEPs count a delivery as failed only when the parcel ends up in the depot again after the last failed delivery attempt. At this point, several additional processes may already have been carried out, which depend on the CEP's general terms and conditions, and the lived practice. Amongst others, the driver would search for a willing neighbor, bring the delivery to a parcel shop in the neighborhood or return the package to the depot to start a new attempt on another day. That way, up to three delivery attempts may take place until the parcel is finally returned to the sender (e.g., Hermes 2020). In all of these cases, it is reasonable to count a delivery attempt as failed since this immediately requires an additional effort. Edwards et al. (2009) report on failed first-time delivery attempts in the magnitude of 10% to 50% where specific delivery times were not prearranged. More recent numbers of Okholm et al. (2013) show an average of 12% and a maximum of more than 50% failed first-time delivery attempts for CEP deliveries in the European Union. It is possible to reduce the probability of failed deliveries by introducing TWs of which customers may choose an appropriate one. However, strict TWs cause significantly higher transportation costs due to less efficient tours (Punakivi and Saranen 2001). This and the CEP's position between sender and receiving customer, often without any contact data of the customer, are the main reasons why TWs are typically not offered in parcel delivery (Wong 2008). Instead of fixed, pre-arranged TWs, customer-individual APs that indicate the probability of a customer being at home can be considered when building delivery schedules (Florio et al. 2018). APs can consist of discrete, non-overlapping TWs with associated probabilities that the customer is available for receiving a delivery within the given TW. APs can be generated in various ways, e.g., by using historical delivery data or based on socio-economic data of the delivery area (Cardenas et al. 2016, van Duin et al. 2016), by electricity consumption data on a household level (Pan et al. 2017) or GPS location data (Praet and Martens 2019). The TW granularity, i.e., the number of TWs within the daily delivery period, should be chosen depending on the available data.

2.2.3 Decision-relevant Costs and Constraints

Costs Considering the distribution process described above, we identify two main cost factors for attended home delivery with failed deliveries: transportation costs for the movement between locations, and expected failed-delivery costs for delivery attempts that may be unsuccessful. Transportation costs arise for each delivery tour to be performed and include the typical costs for traveling between the locations as well as customer-individual service costs to hand over the delivery at the customer's location. Failed-delivery costs depend heavily on the CEP's policy and practice in dealing with failed deliveries. Possible cost-relevant scenarios include, but are not restricted to:

- *Delivery to a neighbor*: Additional service time for the driver.
- *Delivery to a parcel shop*: Time to reload the parcel into the vehicle, transportation costs to reach the parcel shop, fixed fee per parcel when the parcel shop is a partner.
- *Return to the depot & start new delivery attempt next day(s)*: Time to reload the parcel into the vehicle, handling costs at the depot, transportation costs to reach the customer again on the next day(s), additional service time at the second attempt.

Additionally, a second delivery attempt on the same day either within the original tour or performed by another driver who takes over the delivery of failed parcels would be conceivable, but is not a common practice. In all cases, the driver has to leave a notification in the customer's mailbox or trigger an electronic notification for the customer (Hermes 2020). A higher setting of failed-delivery costs may also reflect the increasing importance of CEP's customer satisfaction, as in e-commerce, customers increasingly have the option of selecting the CEP of their choice. As a consequence, failed-delivery costs may include opportunity costs incurred when a customer is lost. The actual costs may further differ between customers due to the availability or unavailability of a parcel shop in the neighborhood of the customer or simply because of greater or less service time required due to local circumstances.

Constraints Various constraints have to be considered within delivery planning of packages to customers' homes. The number of customers on a single delivery tour is restricted by the capacity of the vehicle as well as the length of the delivery period. In the classical VRPTW, waiting times between customers are often necessary to reach all customers within their individual TWs. The problem setting considered here, however, induces waiting times not only to obtain feasible solutions, but also to reduce total costs. The waiting times for all drivers are implicitly limited by the maximum number of vehicles available and the length of the delivery period, as all customers have to be visited within this period by one of the vehicles.

2.2.4 Operational Planning Problem

CEPs engaged in an attended home delivery B2C setting seek to minimize expected costs for the last mile delivery. We consider the single-day operational problem where costs arise from traveling to the customer, serving the customer, and additional costs if the delivery fails. These costs are influenced by decisions regarding (1) the clustering of customers to tours, (2) the delivery sequence on each tour and (3) the selection of delivery TWs. The expected failed-delivery costs depend on the customer's availability within the TW chosen by (3). Service has to start within this chosen TW. In contrast to the VRPTW, the CEP does not have to arrange the tour such that the driver arrives at a specific TW. Instead, arrival is possible throughout the planning horizon, but the customer's availability depends on the TW. Allowing deliveries within the entire delivery period and quantifying the customer's availability via APs enables the CEP to balance between transportation and expected failed-delivery costs.

2.3 Related Literature

The decision problem considered belongs to the class of time-constrained VRPs, in particular VRPs with single or multiple time windows. Section 2.3.1 therefore reviews corresponding time-constrained VRPs first, and Section 2.3.2 the literature that explicitly considers failed deliveries in the context of attended home delivery second. Section 2.3.3 summarizes our findings and specifies the research gap.

2.3.1 Time-constrained Vehicle Routing Literature

The literature on time-constrained VRPs contains models that can be categorized by their assumptions related to the availability of customers within the delivery horizon. With reference to Vidal et al. (2015), we distinguish four problem classes with increasing complexity: the VRP with time windows (VRPTW), the VRP with multiple time windows (VRPMTW), the VRP with soft time windows (VRPSTW) and the VRP with general time windows (VRPGTW). Figure 2.1 visualizes the respective model assumptions of TW constellations and customer availability across the delivery horizon.

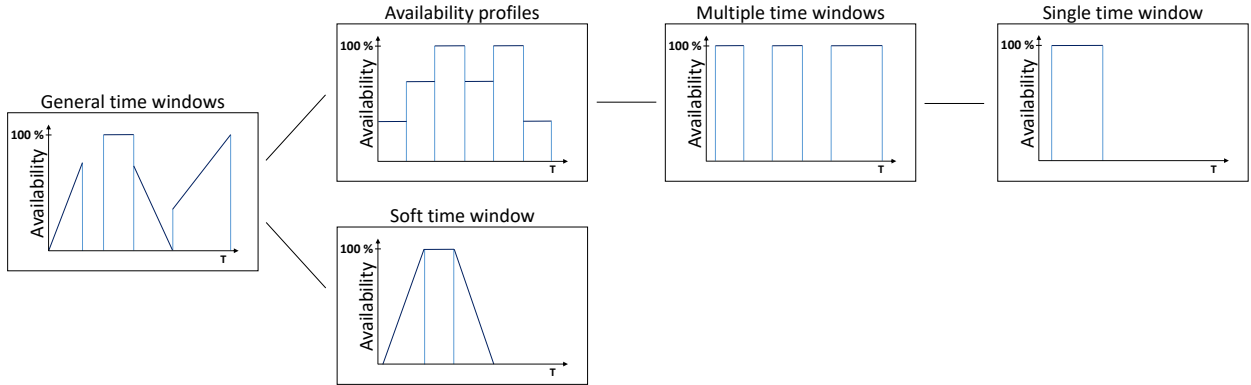


Figure 2.1: Alternative constellations of customer availability across the delivery horizon

The upper branch of Figure 2.1 shows the relation of VRPGTW, VRPAP, VRPMTW, and VRPTW. The VRPAP defines a special case of the VRPGTW with multiple TWs covering the complete delivery horizon and constant failed-delivery costs in every TW. The VRPAP equals the VRPMTW if costs for a failed delivery are infinitely high and TWs are either associated with 0% or 100% customer availability. The VRPAP becomes the VRPTW if there is only one TW with 100% customer availability and all other TWs are associated with 0% customer availability, i.e., there are infinitely high failed-delivery costs for all but one TW. The lower branch of Figure 2.1 shows that the VRPSTW defines another special case of the VRPGTW with a single TW and linearly increasing penalties.

2.3.1.1 VRPTW

Vidal et al. (2013a) present an in-depth review on the diverse set of problem cases discussed in the literature. We however focus on VRPTW publications that present state-of-the-art algorithms and results, which can be used as benchmark for our solution framework proposed. Pisinger and Ropke (2007) present a general heuristic based on the adaptive large neighborhood search (ALNS) for a large class of vehicle routing problems, including the VRPTW. Nagata et al. (2010) present a rather specialized memetic algorithm for the VRPTW. The authors use the Edge Assembly Crossover from Nagata (1997) together with a new penalty function which allows for time warps, i.e., if the vehicle arrives too late at the customer it may use a penalized time warp. Vidal et al. (2013b) build upon the time warp concept and introduce the hybrid genetic search with adaptive diversity control for

VRPs with time features. In contrast to Nagata et al. (2010), their approach works on a large class of time-constrained problems.

2.3.1.2 VRPMTW

De Jong et al. (1996) are the first to formalize the VRPMTW as an MIP. Their objective minimizes total costs, consisting of transportation costs and cost of waiting time. Favaretto et al. (2007) formulate the VRPMTW with periodic constraints. An ant colony optimization algorithm is implemented to solve benchmark instances generated based on VRP instances from the literature, but without considering periodic visits. For the same setting, Belhaiza et al. (2014) develop a hybridized variable neighborhood tabu search heuristic with adaptive memory and solve instances from Favaretto et al. (2007) and newly generated instances based on the VRPTW instances of Solomon (1987), minimizing either travel time or route duration. The latter instances serve as a benchmark for future work. Belhaiza et al. (2017) introduce a hybrid genetic variable neighborhood search for both objectives. Larsen and Pacino (2019) also consider the VRPMTW where the total duration is minimized, as well as a variant where the total travel time is minimized. They implement an ALNS using operators adapted from Ropke and Pisinger (2006) and focus on fast solution evaluations. Belhaiza et al. (2019) present a framework with three different multi-start strategies where they treat solutions as individuals of a genetic population and enhance their VNS with typical destroy-and-repair procedures known from LNS. Hoogeboom et al. (2020) solve the VRPMTW with an adaptive VNS including an exact polynomial time algorithm to determine the optimal service start times of customers in a given route sequence. We refer to this subproblem hereinafter as the optimal start time problem (OSTP). In a similar manner, Schaap et al. (2019) use a LNS with a dynamic programming approach to optimally select a TW for each customer on a route.

2.3.1.3 VRPSTW

Fu et al. (2008) define six different types of soft TWs. Most relevant for and closest to our application are types 1 and 2, where either only late (type 1) or early and late customer visits (type 2) are possible but incur a penalty that is linear to the TW violation. The type 1 variant was first introduced by Taillard et al. (1997) and solved by a tabu search heuristic. The type 2 variant was first considered by Koskosidis et al. (1992) and addressed by solving the clustering problem and the resulting TSPs with soft TWs separately. Like Taillard et al. (1997) they minimize total distance and total penalties simultaneously. Later works on the VRPSTW however use lexicographic objectives. Figliozzi (2010), for instance, first minimizes the number of vehicles, then the number of TWs violated and after that the total distance. The best solution approaches on the commonly used (adapted) Solomon benchmarks for types 1 and 2 that are available currently are Mouthuy et al. (2015), Vidal et al. (2014b) and Kritzing et al. (2017). Mouthuy et al. (2015) use a multistage very large-scale neighborhood search where neighboring solutions are reached not by a single local move but by a sequence of moves in order to achieve better local optima. They also develop a specialized heuristic for the OSTP arising in the VRPSTW. Vidal et al. (2014b)

propose a hybrid genetic search with an efficient local search. Kritzing et al. (2017) adapt the variable neighborhood search method with numerous shaking operators. The latter two approaches are developed to solve a large number of VRP variants including the VRPSTW.

2.3.1.4 VRPGTW

The VRPGTW further extends the VRPMTW and the VRPSTW. It allows customer visits at all times, but at a time-dependent cost defined by a penalty function. In contrast to the typically linear penalties in the case of soft TWs, the penalty functions in the VRPGTW can take any form and also be of a non-convex and discontinuous nature. Nevertheless, piece-wise linearity is assumed in all approaches dealing with the VRPGTW described in literature. Ibaraki et al. (2005) are the first authors who explicitly consider the VRPGTW. They minimize distance costs and penalty costs for TWs used and exceeding vehicle capacity. The problem of determining service start times with minimal penalties for a given route, the OSTP, is identified as a critical subproblem in the VRPGTW since customers can be visited at any time and waiting times between the customers are allowed. They are the first to propose a dynamic programming approach to compute the OSTP. Ibaraki et al. (2008) improve the dynamic programming algorithm of Ibaraki et al. (2005) for the OSTP assuming that the piece-wise linear penalty functions are also convex. Hashimoto et al. (2006) extend the VRPGTW by treating the travel times as time-dependent penalty functions, too, and adapt the dynamic programming algorithm from Ibaraki et al. (2005) to their problem setting.

2.3.2 Failed Delivery Literature

The literature has so far scarcely treated the possibility of a failed delivery in the course of tour planning of attended home deliveries. Pan et al. (2017) use electricity consumption data of customers to reduce failed deliveries in the context of e-groceries. They follow a sequential optimization approach where they first minimize failed-delivery costs by delivering to all customers within their TW with the highest availability and then solving the resulting VRPTW. They show that the approach can decrease the first-time failed-delivery rate by up to 26% in an experiment with 15 customers. However, their approach neglects the potential increase in distance induced by predefined TWs, as well as the potential increasing number of vehicles needed to serve all customers. This so-called ping-pong effect can lead to a significant increase in transportation costs (see, for example, Punakivi et al. 2001). Florio et al. (2018) formulate a delivery problem that also uses APs. In contrast to the VRPAP, they use continuous APs. The model defined minimizes failed deliveries where route duration is restricted. They solve the delivery problem with a branch-and-price algorithm for up to 100 customers. The algorithm leads to optimal solutions if revisits are not allowed and provides a heuristic solution for instances where revisits are possible. They find that using APs for designing routes may result in an average reduction of 34% of failed deliveries. The modeling approach of Florio et al. (2018), however, neglects to include the simultaneous optimization of transportation costs and expected failed-delivery costs. Their modeling approach requires artificial constraints on the additional distance tolerable that have to be defined in advance. Özarık

et al. (2021) present a VRP with discrete customer availability profiles and time-dependent penalty costs, where transportation and expected failed-delivery costs are simultaneously minimized. In contrast to the VRPAP, they neglect vehicle capacities and set the failed-delivery costs to the back and forth distance between the depot and the respective customer instead of a more flexible policy-dependent parameter. The authors propose an ALNS to solve the model formulated, and include a local search procedure to optimize the service start times for given customer sequences, which is similar to the approach of Ibaraki et al. (2005). In their experiments they find that incorporating APs in tour planning reduces total costs by up to 40%.

2.3.3 Summary and Research Gap

To sum up, the problem of failed deliveries in attended home delivery has only rarely been studied in the literature. Existing approaches neglect the effect of increasing distances and the increasing number of vehicles required when introducing TWs or miss features of the operational planning problem, e.g., policy-dependent failed delivery costs and vehicle capacities. An exhaustive modeling approach is needed to conduct research towards numerous open research questions, e.g., the importance of the number of available vehicles, the influence of customer-individual failed-delivery costs and the effects of differently structured APs. Also, the benefit of using the VRPAP instead of existing modeling approaches needs to be clarified. We use our VRPAP modeling approach presented below to address these questions. The VRPAP extends the VRPMTW literature by introducing a valuation of all defined TWs. VRPMTW models assume that the selected TW does not directly affect the objective function. In the VRPAP TWs are valued by availability profiles. The VRPGTW has received only scant attention in the literature so far because it lacks convincing applications. The VRPAP can be seen as a special case of the VRPGTW with constant costs for TWs and has a convincing application in attended home parcel delivery. Our proposed HALNS framework is developed to specifically solve the VRPAP, but is also suitable for solving related time-constrained problem settings.

2.4 Decision Model for Vehicle Routing with Availability Profiles

The VRPAP is defined on a directed graph $G(N, A)$ with node set N and arc set A . Node set N consists of the depot 0 and customers $j \in C$. The arc set is defined as $A = \{(i, j) : i \neq j, i, j \in N\}$. c_{ij}^{trans} denotes the associated transportation costs with each arc. The depot serves as start and end point of all delivery tours. We assume a given fleet of K delivery trucks that states the maximum number of tours. All trucks have the same restricted vehicle capacity Q , while the customer's $j \in C$ demands d_j may be different. Split deliveries are not allowed. The time of a tour consists of the traveling time t_{ij} for driving from one node to the next, of the customer-individual service time S_j , and possibly the waiting time. The maximum route duration is restricted by the delivery period's length D . We divide the delivery period for each customer into $|W|$ discrete, non-overlapping TWs with lower and upper limits e_{jw} and l_{jw} with associated probabilities p_{jw} that customer j is available for receiving a delivery within the given TW w . The customer's individual failed-delivery cost parameter c_j^{failed} denotes the costs that occur if a delivery attempt fails. We are able to represent different policies in dealing with failing deliveries with the same model by setting c_j^{failed} as policy-dependent. The binary decision variable x_{ij} indicates whether arc (i, j) is used. The binary decision variable y_{jw} becomes 1 if TW w is chosen for delivering to customer j . Table 2.1 summarizes the notation.

Table 2.1: Notation

| Sets | |
|-------------------------|--|
| C | Set of customers, $C = \{1, \dots, C \}$ |
| N | Set of nodes, $N = \{0\} \cup C = \{0, \dots, C \}$ |
| W | Set of TWs, $W = \{1, \dots, W \}$ |
| Parameters | |
| c_{ij}^{trans} | Transportation cost from i to j , $i, j \in N$ |
| t_{ij} | Traveling time from i to j , $i, j \in N$ |
| d_j | Demand of customer j , $j \in C$ |
| S_j | Service duration at customer j , $j \in C$ |
| p_{jw} | Availability probability of customer j during TW w , $j \in C$, $w \in W$ |
| c_j^{failed} | Cost of failed delivery attempt for customer j , $j \in C$ |
| e_{jw} | Earliest start of service of TW w for customer j , $j \in C$, $w \in W$ |
| l_{jw} | Latest start of service of TW w for customer j , $j \in C$, $w \in W$ |
| K | Maximum number of vehicles |
| Q | Maximum capacity of a vehicle |
| D | Length of delivery period |
| Decision Variables | |
| x_{ij} | Binary variable indicating whether arc (i, j) is used, $(i, j) \in A$ |
| y_{jw} | Binary variable indicating whether TW w is chosen for customer j , $j \in C$, $w \in W$ |
| s_i | Start time of service at node i , $i \in N$ |
| q_i | Accumulated load until node i , $i \in N$ |

Model VRPAP

$$\text{Minimize } C_{\text{VRPAP}}^{\text{total}} = \sum_{i \in N} \sum_{j \in N} c_{ij}^{\text{trans}} x_{ij} + \sum_{j \in C} \sum_{w \in W} c_j^{\text{failed}} (1 - p_{jw}) y_{jw} \quad (2.1)$$

s.t.

$$\sum_{j \in C} x_{0j} \leq K \quad (2.2)$$

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in C \quad (2.3)$$

$$\sum_{j \in N} x_{ij} = \sum_{j \in N} x_{ji} \quad \forall i \in N \quad (2.4)$$

$$q_j - q_i \geq d_j - Q(1 - x_{ij}) \quad \forall i, j \in C, i \neq j \quad (2.5)$$

$$d_j \leq q_j \leq Q \quad \forall j \in C \quad (2.6)$$

$$s_j + S_j + t_{j0} \leq D \quad \forall j \in C \quad (2.7)$$

$$s_j - s_i \geq (t_{ij} + S_i)x_{ij} - D(1 - x_{ij}) \quad \forall i, j \in C, i \neq j \quad (2.8)$$

$$\sum_{w \in W} e_{jw} y_{jw} \leq s_j \leq \sum_{w \in W} l_{jw} y_{jw} \quad \forall j \in C \quad (2.9)$$

$$\sum_{w \in W} y_{jw} = 1 \quad \forall j \in C \quad (2.10)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (2.11)$$

$$y_{jw} \in \{0, 1\} \quad \forall j \in C, w \in W \quad (2.12)$$

$$s_i, q_i \in \mathbb{R}_0^+ \quad \forall i \in N \quad (2.13)$$

The objective function (2.1) minimizes the total expected costs, which consist of total transportation costs and expected costs of failed delivery attempts. Costs for a delivery that failed arise with the absence probability multiplied by the cost of failed deliveries if the respective TW is chosen. Constraint (2.2) restricts the number of vehicles used. Constraints (2.3) ensure that each customer is visited exactly once. Constraints (2.4) conserve flow. Constraints (2.5) and (2.6) ensure feasibility of loads. Constraints (2.7) ensure that vehicles arrive back at the depot again before the end of the delivery period. Constraints (2.8) guarantee that the service of an immediate successor starts only after traveling and service time starting from the predecessor. Constraints (2.9) ensure that the service must start within the chosen TW. Constraints (2.10) ensure that exactly one TW is chosen for each customer. Constraints (2.11) - (2.13) define the domains of the variables.

The VRPAP generalizes the VRPTW (see Section 2.3.1). This means it is also \mathcal{NP} -hard. Only small instances can be solved by exact approaches. Heuristics are required to solve larger instances. In the following we derive a lower and upper bound to benchmark solutions achieved by our heuristic approach.

Theorem 1. *Let $C_{\text{VRP}}^{\text{trans}}$ denote the resulting costs, when solving the VRPAP ignoring failed-delivery costs, i.e., $c_j^{\text{failed}} = 0, \forall j \in C$. These costs equal the transportation costs when solving the corre-*

sponding VRP. Then, a lower bound for the VRPAP is denoted as follows.

$$C_{\text{VRPAP}}^{\text{lower}} = C_{\text{VRP}}^{\text{trans}} + \sum_{j \in C} c_j^{\text{failed}} (1 - \max(p_{j1}, \dots, p_{j|W|})) \quad (2.14)$$

Proof.

$$\begin{aligned} C_{\text{VRPAP}}^{\text{total}} &= \sum_{i \in N} \sum_{j \in N} c_{ij}^{\text{trans}} x_{ij} + \sum_{j \in C} \sum_{w \in W} c_j^{\text{failed}} (1 - p_{jw}) y_{jw} \\ &\stackrel{(a)}{\geq} \sum_{i \in N} \sum_{j \in N} c_{ij}^{\text{trans}} x_{ij} + \sum_{j \in C} c_j^{\text{failed}} (1 - \max(p_{j1}, \dots, p_{j|W|})) \\ &\stackrel{(b)}{\geq} C_{\text{VRP}}^{\text{trans}} + \sum_{j \in C} c_j^{\text{failed}} (1 - \max(p_{j1}, \dots, p_{j|W|})) = C_{\text{VRPAP}}^{\text{lower}} \end{aligned} \quad (2.15)$$

Inequality (a) states that the TWs with the highest availability probability result in lower or equal costs than the selected TWs in any VRPAP solution. Inequality (b) results from the definition of $C_{\text{VRP}}^{\text{trans}}$ as the the lowest transportation costs, determined by solving the corresponding VRP. \square

An upper bound $C_{\text{VRPAP}}^{\text{upper}}$ for the VRPAP can be derived as follows. We can quantify the sum $C_{\text{VRP}}^{\text{total}}$ consisting of the minimum transportation costs $C_{\text{VRP}}^{\text{trans}}$ and the expected failed-delivery costs for each customer using the associated TWs resulting from solving the VRP. Furthermore, we calculate total costs $C_{\text{VRPTW}}^{\text{total}}$ with the minimum possible failed-delivery costs by solving a VRPTW assuming that the respective TWs with maximum availability apply to each customer. Obviously, $\min(C_{\text{VRP}}^{\text{total}}, C_{\text{VRPTW}}^{\text{total}})$ represents an upper bound $C_{\text{VRPAP}}^{\text{upper}}$ for the VRPAP.

2.5 Hybrid Adaptive Large Neighborhood Search

This section presents a novel hybrid adaptive large neighborhood search (HALNS) for solving the VRPAP. Our HALNS combines elements of an adaptive large neighborhood search (ALNS) as introduced by Ropke and Pisinger (2006) and Pisinger and Ropke (2007), the parallelized ALNS (Mühlbauer and Fontaine 2021) and elements of genetic algorithms, i.e., a population management of individuals (also called solutions), and a crossover phase. The general idea is to maintain a population of individuals that are improved via several ALNSs. Compared to classical ALNSs, these ALNSs use the information available within the population during its optimization procedure. The HALNS relies on this feature as the inner ALNS is optimized for finding reasonably good solutions within very short runtimes as the ALNS is run multiple times. In doing so, we pursue two goals. First, the population-based approach expands the search space, and second, the neighborhood-centred ALNS intensifies the search around promising solutions. Both approaches in combination increase the chance to escape local optima.

Algorithm 4 describes the general structure of the HALNS proposed. We consider an initial population P of size n_P . The initial population of individuals is generated by executing an ALNS n_P times, resulting in potentially several different reasonably good solutions (lines 1-3). The ALNS is then used to crossover individuals (line 6-9), while the number of generations without

Algorithm 4: Hybrid Adaptive Large Neighborhood Search

```

1 while  $|P| < n_P$  do // Generation of Initial Population
2    $s \leftarrow \text{ALNS}()$ 
3    $P \leftarrow P \cup \{s\}$ 
4 while  $\text{gens without Improvement} < \text{gen}^{\text{stop}} \wedge \text{gens} < \text{gen}^{\text{max}}$  do // GA Generations
5    $\hat{s} \leftarrow \text{DetermineBestSolution}(P)$ 
6   while  $i < n_P$  do // Crossover Phase
7      $s \leftarrow P[i]$ 
8      $s \leftarrow \text{ALNS}(s, \hat{s})$ 
9      $P \leftarrow P \cup \{s\}$ 
10   $P \leftarrow \text{SelectSurvivors}(P)$  // Select surviving individuals
    
```

improvement is lower than gen^{stop} or the number of generations has not reached its limit, gen^{max} (line 4). The individuals of a population that are used within the next generation (survivors) are chosen according to their solution quality and their contribution to the diversity of the population (line 10). The proposed ALNS is described in Section 2.5.2. Section 2.5.3 then details the crossover phase. Section 2.5.4 describes the selection logic of survivors.

2.5.1 Solution Representation and Penalized Costs

A solution is represented by a set of routes R , where every customer is covered exactly once by exactly one route. Let r be a route in R , i.e., a sequence of $|r| - 1$ customer visits, which starts at the depot ($r_0 = 0$), ends at the depot ($r_{|r|} = 0$) and covers customers in between ($r_{(1, \dots, |r|-1)} \neq 0$). Following the idea of Vidal et al. (2013b), we allow non-feasible solutions in the search space by penalizing the degree of infeasibility with an additional weight ω , i.e., a penalty cost factor, in the cost function. We then combine the costs for transportation C_r^{trans} and failed delivery C_r^{failed} with possible penalty costs for an overloaded vehicle P_r^{Load} and TW violations P_r^{TW} . The adapted entire costs of a route r , $f(r)$, are hereafter denoted as penalized costs.

$$f(r) = C_r^{\text{trans}} + C_r^{\text{failed}} + \omega(P_r^{\text{Load}} + P_r^{\text{TW}}) \quad (2.16)$$

Let $p_j^{\text{available}}$ be the availability probability resulting from choosing a TW for customer j in r . Furthermore, let P_{ir} be the lateness resulting from being too late for the designated TW w , or being too late back at the depot for node i in r , which is defined as follows.

$$P_{ir} = \begin{cases} \max\{s_i - l_{iw}, 0\} & i \in r, i \neq 0 \\ \max\{s_{r_{|r|-1}} + S_{r_{|r|-1}} + t_{r_{|r|-1}, i} - D, 0\} & i = 0 \end{cases} \quad (2.17)$$

The individual terms of route r are then calculated as follows:

- Transportation costs: $C_r^{\text{trans}} = \sum_{i=0}^{|r|-1} c_{r_i, r_{(i+1)}}^{\text{trans}}$
- Failed-delivery costs: $C_r^{\text{failed}} = \sum_{j=1}^{|r|-1} c_j^{\text{failed}} (1 - p_j^{\text{available}})$
- Excess truck load: $P_r^{\text{Load}} = \max\{0, \sum_{j=1}^{|r|-1} d_j - Q\}$

- TW lateness: $P_{ir}^{\text{TW}} = \sum_{i=0}^{|r|} P_{ir}$

The penalized cost $f(s)$ of a solution s is then the sum of the penalized costs of all routes in R , i.e., $f(s) = \sum_{r \in R} f(r)$.

2.5.2 ALNS Algorithm

Algorithm 5: ALNS algorithm in simulated annealing framework

Input : Starting solution s , global best solution \hat{s} , temperature γ
Output: best solution s^*

```

1  $s^* \leftarrow s$ 
2 while Iterations without improvement <  $it^{\text{limit}}$  do
3   ChooseOperators()
4    $C_R^C \leftarrow \text{getRemovalCandidates}(s, \hat{s})$ 
5    $(s^{\text{new}}, C_R) \leftarrow \text{Remove}(s, C_R^C)$ 
6    $C_R \leftarrow \text{sort}(C_R)$ 
7    $s^{\text{new}} \leftarrow \text{Insert}(s^{\text{new}}, C_R)$ 
8   if  $f(s^{\text{new}}) < f(s^*)$  then
9      $(s, s^*) \leftarrow s^{\text{new}}$ 
10    if  $f(s^*) < f(\hat{s})$  then
11       $\hat{s} \leftarrow s^*$ 
12    else if  $\text{accept}(f(s^{\text{new}}), f(s^*), \text{Temp})$  then
13       $s \leftarrow s^{\text{new}}$ 
14     $\gamma \leftarrow \beta \cdot \gamma$ 
15    UpdateWeights()
    
```

The ALNS takes the current solution s , the global best solution \hat{s} and the starting temperature γ as input. Initially, the local best solution s^* is set, and the ALNS generates new solutions by iteratively removing and then inserting customers from and into the current solution. In each iteration, a removal operator (see Section 2.5.2.3) and an insertion operator (see Section 2.5.2.5) are randomly chosen (line 3). Hereby, the selection probability depends on the historic performance of the respective operator. Different from classical ALNS implementations, a set of customers that are candidates for removal C_R^C is generated by comparing the current solution s and the global best solution \hat{s} (line 4, see Section 2.5.2.2). The removal operator chosen then removes some or all customers of set C_R^C from the solution (line 5). Additionally, data collected during the search determines the order in which the removed customers C_R are inserted (line 6). The determination of the insertion order (see Section 2.5.2.4) is an extension to existing ALNS implementations that either randomly insert customers or determine the order during the insertion process (e.g., regret-insertion Ropke and Pisinger 2006). Afterwards, the insertion operator chosen inserts the set of previously removed and sorted customers into the solution (line 7). We use a simulated annealing acceptance criterion (see Section 2.5.2.6) with cooling rate β to avoid getting stuck in a local optimum (lines 8-14), like other ALNS implementations (e.g., Ropke and Pisinger 2006, Larsen and Pacino 2019). Finally, the weights are updated (see Section 2.5.2.7) according to the performance of the operators (line 15). The ALNS ends after a predefined number of iterations without any improvements, it^{limit} .

Note that the HALNS generates the initial solution in the first generation by iteratively applying the *Best Route Insertion Operator* (see Section 2.5.2.5) until every customer is served. In the

following generations, the initial solution of an ALNS phase is chosen from the population of solutions.

2.5.2.1 Information Collection

The insertion operators collect three types of information when inserting customers. This information is used during the subsequent removal and insertion operations.

- The historic penalized costs C_j^{hist} is increased on every insertion of customer j by $\Delta_j = f(s_1) - f(s_0)$, where s_1 represents the solution after inserting customer j and s_0 the solution directly before, i.e., without customer j . C_j^{hist} measures the accumulated increase in costs when inserting customer j . These costs approximate the additional cost effort when inserting customer j into a given solution.
- C_j^{min} quantifies the minimum insertion costs of customer j found so far. It is updated every time a new insertion position with lower costs is found, i.e., every time $\Delta_j < C_j^{\text{min}}$.
- n_j counts the number of times customer j is reinserted.

2.5.2.2 Determination of Removal Candidates

The set of candidates to be removed, C_R^C , defines a newly designed feature in the ALNS proposed. The set is determined by comparing the current solution s with the global best solution \hat{s} . We use two variants for selecting which customers are added to set C_R^C . One of these two alternatives is randomly chosen in every iteration. The probability depends on the performance and is adapted during the search (see Section 2.5.2.7).

Sequence Comparison A customer is added to C_R^C if its successor in s differs from its successor in \hat{s} . This indicates that the customer is oddly placed compared to the position in the best solution found so far. Customers who do not differ from their successors in \hat{s} are nevertheless added to set C_R^C with a probability p^{binom} for diversification purposes. We use this procedure since we expect that a certain number of edges in the best solution found so far will be similar to the optimal solution. We therefore combine the information from different solutions in a less random fashion than in a crossover but still include the information of the whole population.

Time Window Comparison A second variant of the procedure to determine C_R^C is implemented where the TW assignment of s and \hat{s} is compared instead of the immediate successor. To be more specific, a customer will be added to C_R^C if the customer has a different TW assigned in s and \hat{s} .

2.5.2.3 Removal Operators

The maximum number of customers to be removed, q^{binom} , is sampled from a binomial distribution in every iteration (Voigt and Kuhn 2021). The binomial distribution takes two parameters, namely the sample size that we set to $|C|$ and the probability p^{binom} that is adapted during one ALNS run. The dynamic adaptation of this parameter is a new component that we use to cope with

different instance characteristics (e.g., APs, geographical distributions, vehicle capacities). In some instances only a few customers should be removed, reducing the ALNS to a more local search. In other instances, a high number of customers have to be removed in order to change the solution. The number of customers that are actually removed is expressed by $q^r = \min\{q^{\text{binom}}, |C_R^C|\}$. The ALNS uses four known removal operators (Ropke and Pisinger 2006), which we adapt to our problem setting. We have chosen simple yet efficient operators and further optimize them to find reasonable solutions within short runtimes.

Random Removal The *Random Removal Operator* randomly selects q^r customers out of set C_R^C and removes these customers from the current solution. This operator is fast and fosters diversification.

Historic Cost Removal The *Historic Cost Removal Operator* uses the historic information gained during the insertion procedures (see Section 2.5.2.1) and removes q^r customers from set C_R^C in decreasing order of average historic penalized costs, i.e., in decreasing order of $\frac{C_j^{\text{hist}}}{n_j}$. The operator thus uses average historic penalized cost rather than just the costs encountered during the previous insertion phase. Otherwise the operator would show the tendency of removing the customers who have been inserted at the end of the previous insertion step, because these customers are constrained most concerning available insertion possibilities and therefore naturally show higher costs.

Worst Cost Removal The *Worst Cost Removal Operator* uses the historic information collected during insertion, too. The operator calculates the change in transportation costs, Δ_j , if customer j is removed from the current solution. It then compares this cost difference to the minimum encountered penalized cost during the search C_j^{min} . Customers are removed in descending order of $\Delta_j - C_j^{\text{min}}$, i.e., customers with a high difference in costs, who therefore seem to be oddly placed in the solution, are removed. The operator is similar to the well-known *Worst Removal Operator*, but overcomes the tendency of removing the same customer all over again only because the customer is for example far away from every other customer and therefore naturally increases the costs the most.

Shaw Removal The *Shaw Removal Operator* removes customers from the solution that are similar to each other. The first customer is removed with the *Random Removal Operator*. The following requests are chosen from C_R^C in increasing order of relatedness. The relatedness $Rel(c_1, c_2)$ of two customers c_1 and c_2 is measured by the distance of both customers and the similarity of their earliest completion times $E(j)$ in s . The lower the $Rel(c_1, c_2)$, the more related are customers c_1 and c_2 .

$$Rel(c_1, c_2) = \frac{c_{c_1, c_2}}{\max_{i, j \in N} c_{ij}} + \frac{|E(c_1) - E(c_2)|}{l_0} \quad (2.18)$$

Customers with a lower $Rel(c_1, c_2)$ are removed first.

2.5.2.4 Determination of Insertion Order

At the beginning of the insertion process, the set of removed customers, C_R is sorted in descending order of average historic costs $\frac{C_j^{\text{hist}}}{n_j}$. Customers who are hard to insert without incurring high penalized costs are inserted earlier and therefore have a greater chance of being inserted at an appropriate position. The later a customer is inserted into a tour, the more difficult it is to find a position that will increase the total costs at a moderate level only. Customers with low average historic costs should therefore be inserted at a later stage.

2.5.2.5 Insertion Operators

Two decisions have to be made when inserting a customer into a route: the position within the sequence of already assigned customers and the TW when the customer should be served. The selection of an appropriate TW can however influence the sequence and the TWs of already assigned customers. Within the ALNS, we therefore restrict the insertion operators to only allow the insertion of customers if TWs and sequences of customers on the route are not shifted. Based on this general strategy we define two variants of the *Best Insertion Operator*. Both variants iterate across all routes to find the insertion position that increases the cost least. Note that we again avoid the use of computationally intensive operators such as k-regret that are otherwise widely used in the ALNS literature (e.g., Larsen and Pacino 2019).

Best Insertion: First Feasible TW For a given insertion position in a route, the *Best Insertion: First Feasible TW Operator* uses the earliest feasible TW. The feasibility of a selected TW can easily be checked in constant time by using the concept of reoptimization by concatenation of sequences (Vidal et al. 2013b). In the worst case, each and every TW has to be checked for every insertion position on a route, resulting in a worst-case complexity for the insertion of one customer of $\mathcal{O}(n|TW|)$. Here, n denotes the current number of customers on a route and $|TW|$ indicates the number of TWs.

Best Insertion: Best Feasible TW The *Best Insertion: Best Feasible TW Operator* uses the TW that leads to the lowest penalized costs (compared to the first feasible TW) without shifting the sequence or the TW assignment of all other customers on that route. This operator has an average and worst-case complexity of $\mathcal{O}(n|TW|)$.

2.5.2.6 Simulated Annealing

The initial temperature is determined for every instance with $\gamma = -\frac{\overline{\Delta E}}{\ln(\chi_0)}$ using the formula from Johnson et al. (1989) as cited in Ben-Ameur (2004). $\overline{\Delta E}$ estimates strictly positive cost increases and χ_0 expresses the probability of accepting a worse solution. We execute n_0 iterations of the ALNS to generate the transitions. The temperature γ thus determined is reduced by β after each iteration.

2.5.2.7 Adaptivity

The score of an operator is increased by either σ_1 , σ_2 or σ_3 . It is increased by σ_1 if a new best solution is found, by σ_2 if a previously unknown solution with lower costs than the current solution is found, or by σ_3 if the solution has higher costs but is accepted through the simulated annealing procedure. The probability of choosing removal and insertion operators depends on the scores and is calculated as in Ropke and Pisinger (2006).

2.5.3 Crossover Phase

The crossover phase uses the ALNS described in the previous section. In each generation, the ALNSs use the individuals in the current population one by one as a starting solution. The determination of the removal candidates is influenced by the global best solution (see Section 2.5.2.2). In a genetic algorithm, the idea of a crossover is to combine two individuals hoping to maintain good segments of a solution. In contrast, we try to achieve the same effect by reducing the probability of replacing well-placed customers. This procedure is based on the consideration that the optimal solution is likely to have some similarities with the solutions generated by ALNS and even more similarities with the best solution of the population.

2.5.4 Selection of Survivors

Surviving individuals are determined on the basis of total costs and the contribution of each individual to the diversity of the population similar to the diversity management of Vidal et al. (2012). All individuals of population P are placed in increasing order of total costs and a rank is assigned to the individual i , $Rank_i^{Costs}$, e.g., the individual with the lowest cost gets $Rank_i^{Costs} = 0$, the individual with the highest cost $Rank_i^{Costs} = |P| - 1$.

The diversity of an individual i is calculated by the hamming distance to all other individuals in the population based on the successor of nodes, $Succ^i$.

$$Hamming^{ind} = \sum_{i \in Population} \sum_{j \in C} Succ_j^{ind} \neq Succ_j^i \quad (2.19)$$

The population is placed in decreasing order of the hamming distance and a diversity rank is assigned to the individual $Rank_i^{Diversity}$. The overall rank is calculated by summing up the cost and diversity rank, $Rank_i^{Total} = Rank_i^{Costs} + Rank_i^{Diversity}$. Individuals with the n^P lowest total ranks are chosen for the next generation. Individuals with the $4 \cdot n^P$ lowest total ranks survive, meaning that an individual can be used in later generations, even if not used during previous generations. Lastly, after each fifth generation new individuals are generated in the same manner as the first generation to further enhance diversity.

2.6 Numerical Experiments

This section presents our numerical experiments based on simulated and empirical data. The HALNS is implemented in *C++* and run on an AMD Ryzen 9 3900X with 32GB RAM. Sec-

tion 2.6.1 details how we generated VRPAP instances. Section 2.6.2 evaluates the performance of the HALNS on related time-constrained VRP instances from literature and VRPAP instances generated, and additionally analyzes HALNS components. Section 2.6.3 generates managerial insights. Section 2.6.4 presents a case study with real APs. Table 2.2 gives an overview of experiments and data sets.

Table 2.2: Experiments and data sets

| Section | Experiments and purpose | Data set used |
|---------|---|--|
| 6.1 | VRPAP Instance Generation | Solomon (1987), Florio et al. (2018) |
| 6.2 | Performance Evaluation | |
| 6.2.1 | Benchmark for Time-Constrained VRPs | Solomon (1987), Belhaiza et al. (2014) |
| 6.2.2 | Benchmark for VRPAP Instances | Synthetic VRPAP instances generated in 6.1 |
| 6.2.3 | Analysis of Algorithmic Components | Synthetic VRPAP instances generated in 6.1 |
| 6.3 | Sensitivity Analysis and Managerial Insights | Synthetic VRPAP instances generated in 6.1 |
| 6.4 | Case Study with Empirical Availability Profiles | Solomon (1987), Time Use Survey Data (Gershuny and Sullivan 2017) |

2.6.1 VRPAP Instance Generation

We use instances for the VRPTW by Solomon (1987) and replace the TWs by different APs derived from Florio et al. (2018). Every profile consists of 10 TWs and has an average availability of 50% (see Table 2.3). The A-profile has a peak during midday. The V-profile has two peaks, one in the morning and one in the evening. The W-profile has an additional peak during midday and can therefore be considered a combination of A and V profiles. The M-profile is the opposite of the W-profile, i.e., when the availability of the W-profile is high, the availability of the M-profile is low. We generate instances based on these profiles where all customers of one instance have the same profile (A, V, W or M), and instances with a combination of profiles (AV, WM, and AVWM). In the latter case, customers of one instance are randomly assigned one of the respective profiles in equal proportions. We combine the seven different pure and mixed APs (A, V, W, M, AW, WM and AVWM) with six geographical distributions (R1, R2, C1, C2, RC1 and RC2) of customers from Solomon (1987), resulting in 42 instances.

2.6.2 Performance Evaluation

Section 2.6.2.1 evaluates the performance of the HALNS on special cases of the VRPAP, namely the VRPTW and the VRPMTW, as well as the more complex VRPSTW. Section 2.6.2.2 compares

Table 2.3: Availability profiles

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| A | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 | 0.9 | 0.7 | 0.5 | 0.3 | 0.1 |
| V | 0.9 | 0.7 | 0.5 | 0.3 | 0.1 | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
| W | 0.8 | 0.4 | 0.1 | 0.4 | 0.8 | 0.8 | 0.4 | 0.1 | 0.4 | 0.8 |
| M | 0.2 | 0.6 | 0.9 | 0.6 | 0.2 | 0.2 | 0.6 | 0.9 | 0.6 | 0.2 |

the HALNS solution for VRPAP instances to a lower and upper bound. Section 2.6.2.3 evaluates the importance of specific algorithmic components of the HALNS. We use the identical set of algorithmic parameters (see Appendix A.1) that has been selected after preliminary testing for all experiments.

2.6.2.1 Benchmark for Time-Constrained VRPs

Tables 2.4 to 2.7 summarize the results. Column *BKS* shows the (previous) best-known solution averaged over distribution types, column *Best X* shows the averaged best results obtained during 5 or 10 runs, and column *Avg X* demonstrates the arithmetic mean over 5 or 10 runs (combined by *Best/Avg* if only one run has been executed). Row Σ presents the cumulated objective values over the whole instance set, row *Avg gap* represents the average gap compared to the (previous) BKS, and row *#BKS* the number of BKS found. Furthermore, row *Avg T* shows the arithmetic mean of the runtime over 5 or 10 runs. We also specify the processors used and their passmark single thread rating (<https://www.cpubenchmark.net/>) in order to make the runtimes comparable.

VRPTW Benchmark Table 2.4 shows that the performance of the HALNS is comparable to well-performing algorithms for the VRPTW from Nagata et al. (2010) and Vidal et al. (2013b). Figure A.1 in Appendix A.2.1 additionally demonstrates that the HALNS also clearly outperforms the sole implementation of the ALNS by Pisinger and Ropke (2007) for smaller runtimes. This indicates the value of the hybridization of the ALNS.

Table 2.4: Summarized results for Solomon (1987) VRPTW instances

| | BKS | Pisinger | Nagata | Vidal | | HALNS | |
|----------|---------|----------------|----------------|--------------|---------|---------------|---------|
| | | Best 10 | Best 5 | Best 5 | Avg 5 | Best 5 | Avg 5 |
| R1 | 1210.34 | 1212.39 | 1210.34 | 1210.69 | 1211.49 | 1210.35 | 1210.93 |
| R2 | 951.03 | 957.72 | 951.03 | 951.51 | 952.05 | 951.03 | 951.77 |
| C1 | 828.38 | 828.38 | 828.38 | 828.38 | 828.38 | 828.38 | 828.38 |
| C2 | 589.86 | 589.86 | 589.86 | 589.86 | 589.86 | 589.86 | 589.86 |
| RC1 | 1384.17 | 1385.78 | 1384.17 | 1384.17 | 1384.81 | 1384.17 | 1384.18 |
| RC2 | 1119.24 | 1123.49 | 1119.24 | 1119.24 | 1119.4 | 1119.34 | 1119.43 |
| Σ | 57187 | 57332 | 57187 | 57196 | 57218 | 57188 | 57204 |
| Avg T | | 150s | 300s | 161s | | 155s | |
| CPU | | Pentium 4 3GHz | Opteron 2.4GHz | Xeon 2.93GHz | | Ryzen 9 3900X | |
| Passmark | | 561 | 430 | 1418 | | 2731 | |

VRPMTW Benchmark Table 2.5 shows that the HALNS achieves a similar best gap to that of the ALNS of Larsen and Pacino (2019), which is the state-of-the-art approach for the VRPMTW. However, the HALNS outperforms the ALNS in terms of average performance. The HALNS produces 28 BKS out of 48 instances, including 12 newly found BKS. Note that the HALNS already produces reasonably good solutions at much shorter runtimes, as shown in Figure A.2 in Appendix A.2.2.

Table 2.5: Summarized results for Belhaiza et al. (2014) VRPMTW instances

| BKS | Belhaiza 2014 | | Belhaiza 2017 | Larsen | | Schaap | | Hoogeboom | HALNS | | |
|----------|---------------|--------|---------------|----------|--------|-----------|--------|-----------|---------------|-------------|--------|
| | Best 10 | Avg 10 | Best/Avg | Best 10 | Avg 10 | Best 10 | Avg 10 | Best/Avg | Best 10 | Avg 10 | |
| R1 | 2727.8 | 2740.8 | 2755.1 | 2732.3 | 2728.9 | 2731.1 | 2738.6 | 2773.3 | 2738.1 | 2729.5 | 2731.3 |
| R2 | 2674.3 | 2810.7 | 2826.4 | 2793.9 | 2684.7 | 2687.3 | 2691 | 2702.1 | 2679.4 | 2683.5 | 2683.9 |
| C1 | 3217.6 | 3297.3 | 3315.4 | 3278.2 | 3227.6 | 3256.7 | 3243.4 | 3266.3 | 3254.5 | 3231.1 | 3257.3 |
| C2 | 4156.2 | 4192.7 | 4216.5 | 4169 | 4156.2 | 4166.7 | 4163.9 | 4182 | 4177.8 | 4158 | 4162.9 |
| RC1 | 3218.4 | 3244.4 | 3258 | 3229.3 | 3218.4 | 3220.7 | 3222.6 | 3241 | 3235.1 | 3218.2 | 3219.3 |
| RC2 | 2730 | 2899.7 | 2919.2 | 2879 | 2731.8 | 2789.1 | 2732.6 | 2845.4 | 2875.5 | 2727.7 | 2728.4 |
| Σ | 149795 | 153484 | 154324 | 152653 | 149981 | 150812 | 150337 | 152082 | 151683 | 149983 | 150265 |
| Avg gap | 0.00% | 2.60% | 3.16% | 2.04% | 0.13% | 0.70% | 0.37% | 1.60% | 1.33% | 0.13% | 0.31% |
| # BKS | 36/48 | 2 | | 8 | 27 | | 5 | | 5 | 28 (12 new) | |
| Avg T | 64s | | 81s | 600s | | 185s | | 113s | 629s | | |
| CPU | i5 3.3GHz | | i5 3.3GHz | i7-4790K | | i7 3.7GHz | | i7 4GHz | Ryzen 9 3900X | | |
| Passmark | 1704 | | 1704 | 2469 | | 2776 | | 2469 | 2731 | | |

VRPSTW Benchmark For the VRPSTW, we focus on the type 1 (only lateness considered) and type 2 (earliness and lateness considered) variants of the VRPSTW with $\alpha = 1$ (see, Fu et al. 2008). The parameter α sets the amount of linear penalty depending on the earliness/lateness. Setting $\alpha = 1$ means that one time unit of earliness/lateness equals one unit of transportation costs. While the VRPTW and the VRPMTW are special cases of the VRPAP and can be directly solved by the HALNS, the VRPSTW constitutes an extension to the VRPAP. The HALNS therefore needs some adaptations for solving VRPSTW variants. For the case of type 1 we need to adapt the objective function and allow solutions where customers are delivered after their desired TW. For type 2 we additionally need to adapt how operators determine the start of service for customers on a route. We replace every TW by two artificial TWs (with the same earliest and latest start times as the original single one), from which the HALNS may choose one. The first TW only allows earliness, the second TW only allows lateness. If the first TW is chosen, the customer will be served as early as possible even if this results in an earliness penalty. In contrast, if the second TW is chosen, the service does not start before the earliest start time of the TW, which means the vehicle has to wait if it arrives too early. This allows the use of the same operators as before to approximate the earliness/lateness. Note that penalties for earliness/lateness during the insertion phase are only valid for the customer, not for the overall route. The overall route is only evaluated after every customer is inserted.

Tables 2.6 and 2.7 show the summarized results. For type 1 instances, the HALNS achieves results similar to the currently best approach from Vidal et al. (2014b). For type 2 instances, the HALNS reveals a better average gap and finds 8 new BKS. This demonstrates the flexibility of the HALNS. Detailed results may be found in Appendix A.2.3.

2.6.2.2 Benchmark for VRPAP Instances

Table 2.8 shows summarized results of the HALNS algorithm when solving the 42 VRPAP instances generated (see Section 2.6.1). As a comparison we use the lower and upper bound described in Section 2.4 as benchmark. Column *Instance Group* aggregates the results by APs, as there are no significant differences between geographical distributions (see Table A.6 in Appendix A.2.4). The next three columns show the lowest possible distances costs, achieved by solving a VRP denoted by C_{VRP}^{trans} , the lowest possible failed-delivery costs C_{VRPTW}^{failed} , and the lower bound, derived by the summation of both terms before, denoted by C_{VRPAP}^{lower} . Column *Best 10* and *Avg 10*, show the best

Table 2.6: Summarized results for VRPSTW instances type 1 (only lateness), $\alpha = 1$

| BKS | | Kritzinger | | Vidal | | HALNS | |
|----------|----------|--------------|----------|----------------|----------|---------------|----------|
| | | Best 10 | Avg 10 | Best 10 | Avg 10 | Best 10 | Avg 10 |
| C1 | 828.38 | 828.38 | 828.38 | 828.38 | 828.38 | 828.38 | 828.38 |
| C2 | 589.86 | 589.86 | 589.86 | 589.86 | 589.86 | 589.86 | 589.86 |
| R1 | 1170.11 | 1171.37 | 1174.9 | 1170.16 | 1171.11 | 1170.35 | 1170.73 |
| R2 | 946.17 | 952.39 | 972.44 | 946.17 | 947.35 | 946.68 | 946.82 |
| RC1 | 1313.9 | 1313.9 | 1315.34 | 1313.9 | 1314.04 | 1314.26 | 1314.94 |
| RC2 | 1106.61 | 1108.26 | 1114.89 | 1106.61 | 1107.37 | 1106.61 | 1106.8 |
| Σ | 55987.56 | 56084.33 | 56411.68 | 55988.16 | 56019.79 | 55999.01 | 56012.04 |
| Avg gap | 0.00% | 0.19% | 0.81% | 0.00% | 0.06% | 0.02% | 0.04% |
| # BKS | 56/56 | 36 | | 55 | | 50 | |
| Avg T | | 600s | | 349s | | 156s | |
| CPU | | Xeon E7-8837 | | Opteron 2.2GHz | | Ryzen 9 3900X | |
| Passmark | | 1124 | | 445 | | 2731 | |

Table 2.7: Summarized results for VRPSTW instances type 2 (earliness and lateness), $\alpha = 1$

| BKS | | Vidal | | HALNS | |
|----------|---------|----------------|----------|---------------|----------|
| | | Best 10 | Avg 10 | Best 10 | Avg 10 |
| C1 | 828.38 | 828.38 | 828.38 | 828.38 | 828.38 |
| C2 | 589.86 | 589.86 | 589.86 | 589.86 | 589.86 |
| R1 | 1164.86 | 1164.86 | 1167.73 | 1167.38 | 1169.66 |
| R2 | 949.87 | 949.87 | 957.44 | 945.99 | 946.78 |
| RC1 | 1304.23 | 1304.23 | 1304.41 | 1309.14 | 1310.21 |
| RC2 | 1106.43 | 1106.43 | 1108.42 | 1105.98 | 1106.09 |
| Σ | 55886.4 | 55886.4 | 56021.42 | 55909.8 | 55955.17 |
| Avg gap | 0.00% | 0.00% | 0.26% | 0.00% | 0.07% |
| # BKS | 48/56 | 48 | | 31 (8 new) | |
| Avg T | | 1797s | | 159s | |
| CPU | | Opteron 2.2GHz | | Ryzen 9 3900X | |
| Passmark | | 445 | | 2731 | |

and average result of ten HALNS runs. Column C_{VRPAP}^{upper} signifies the upper bound. The next two columns Δ^{lb} and Δ^{ub} represent the percentage gaps to the lower and upper bound, respectively. Lastly, column *Avg T [s]* shows the average runtime of ten runs in seconds.

Compared to the upper bound procedure, the HALNS systematically makes use of APs and therefore finds a balance between transportation and expected failed-delivery costs. This means the HALNS is able to reduce expected costs if information on the AP is at hand. The lower bound is quite weak if the APs are homogenous and only few peaks exist (A, V, M), but becomes slightly better if the APs are more diverse or more peaks exist, i.e., the gap between the objective value obtained by the HALNS and the lower bound becomes smaller. This result is intuitive, because diverse APs (AV, WM, AVWM) and profiles with more peaks (W) offer more flexibility for altering the service time and thereby reducing expected failed-delivery costs without increasing the distance costs.

Table 2.8: Summarized results for VRPAP instances

| Instance Group (R101, ..., RC201) | C_{VRP}^{trans} | C_{VRPTW}^{failed} | C_{VRPAP}^{lower} | Best 10 | Avg 10 | C_{VRPAP}^{upper} | Δ^{lb} | Δ^{ub} | Avg T [s] |
|--------------------------------------|-------------------|----------------------|---------------------|---------|---------|---------------------|---------------|---------------|-----------|
| A | 762.88 | 76.29 | 839.16 | 1070.67 | 1074.72 | 1094.74 | 28% | -2% | 955 |
| V | 762.88 | 76.29 | 839.16 | 1115.71 | 1118.6 | 1192.98 | 33% | -7% | 908 |
| W | 762.88 | 152.58 | 915.45 | 1090.45 | 1094.82 | 1157.51 | 19% | -6% | 973 |
| M | 762.88 | 76.29 | 839.16 | 1068.39 | 1071.04 | 1130.45 | 28% | -5% | 891 |
| AV | 762.88 | 76.29 | 839.16 | 1064.36 | 1067.56 | 1134.55 | 27% | -6% | 982 |
| WM | 762.88 | 114.43 | 877.31 | 1028.71 | 1031.23 | 1140.28 | 18% | -10% | 947 |
| AVWM | 762.88 | 95.36 | 858.23 | 1049.22 | 1052.68 | 1140.41 | 23% | -8% | 943 |
| Avg | 762.88 | 95.36 | 858.23 | 1069.64 | 1072.95 | 1141.56 | 25% | -6% | 943 |

2.6.2.3 Analysis of Algorithmic Components

In order to gain insights into the importance of algorithmic components, we analyze the performance change of the HALNS when selected components are deactivated (−) or activated (+). The following configurations are tested.

HALNS: The baseline HALNS as described in Section 2.5.

(−) **Simulated annealing:** No simulated annealing within the ALNS (see Section 2.5.2).

(−) **Crossover:** No crossover (see Section 2.5.3).

(−) **Removal candidates:** No preselection of customers to be removed (see Section 2.5.2.2).

(−) **Insertion order:** No ordering of customers before inserting (see Section 2.5.2.4).

(+) **Local improvement:** Additional improvement procedure executed on the TW assignment of every new best solution found during the search based on the dynamic programming approach of Ibaraki et al. (2005) for solving the OSTP. The approach allows the determination of the optimal (i.e., penalty- and failed-delivery cost-minimal) service start times and respective TWs for a given sequence of customers on a route. Note that this sequence is retained throughout the algorithm and service start times are only altered by making use of the available waiting time in the tour.

We use the VRPAP instances from Table 2.8 and run each configuration ten times. Table 2.9 shows the average of the best of ten runs in Column *Best 10*, the average in Column *Avg 10* and the average runtime in seconds in Column *Avg T [s]*. All components, except *insertion order* and *local improvement* improve the best gap significantly. Also, all components (again, except for *local improvement*) further stabilize the average gap, resulting in a robust solution approach and additionally reduce the average runtime significantly. The findings are consistent over runtime (see Figure A.3 in Appendix A.3). To summarize, only the component *local improvement* is not beneficial for solving VRPAP instances, which means that the HALNS finds the optimal TW assignment without a dedicated algorithmic component. The slightly worse average performance indicates that results can even get worse when the TW assignment is optimized prematurely and the search gets stuck in a local optimum.

Table 2.9: Influence of specific components on HALNS performance

| Algorithm | Best 10 | Avg 10 | Avg T [s] |
|-------------------------|---------|---------|-----------|
| HALNS | 1069.64 | 1072.95 | 943 |
| (-) Simulated annealing | 1083.94 | 1101.33 | 1520 |
| (-) Crossover | 1073.73 | 1080.06 | 1227 |
| (-) Removal candidates | 1072.34 | 1079.04 | 1107 |
| (-) Insertion order | 1069.05 | 1073.53 | 1008 |
| (+) Local Improvement | 1069.41 | 1073.63 | 972 |

2.6.3 Sensitivity Analysis and Managerial Insights

In the following we generate insights into the benefits and implications of the VRPAP. Section 2.6.3.1 evaluates the cost savings potential of the VRPAP against current approaches. Section 2.6.3.2 analyzes which APs promise the highest cost reductions. Section 2.6.3.3 demonstrates the influence of c^{failed} on the solution structure. Section 2.6.3.4 analyzes the tradeoff between distance costs and failed-delivery rate. Section 2.6.3.5 examines the effect of applying a back-up policy when a delivery fails.

2.6.3.1 Cost Savings Potential

This section determines the cost savings achievable when using the VRPAP compared to applying myopic VRPMTWs with varying numbers of the most likely TWs. The VRPMTW includes the VRP (all TWs allowed) and the VRPTW (one most likely TW allowed) as special cases. This analysis quantifies the failed-delivery cost parameter c^{failed} as a multiple of $\overline{C^{\text{VRP}}}$. The cost value $\overline{C^{\text{VRP}}}$ denotes the average transportation costs per customer neglecting TWs, i.e., the cost resulting from solving the VRP divided by the number of customers served. This assumes that the CEP follows a policy in which customers whose delivery has failed are served at average delivery costs the next day.

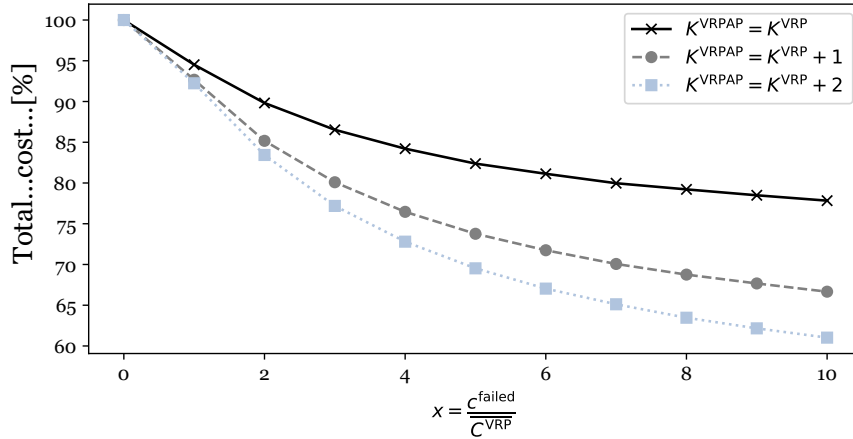
We recalculate a reduced set of 32 instances from Table 2.8 for this experiment, as the approach with the most likely TW (i.e., the VRPTW) has no feasible solution for ten instances because this specific TW cannot be reached in time when originating from the depot. The upper section of Table 2.10 shows the average results when solved as VRPAP and as VRPMTW with 1, 3, 5, or 10 hard TWs that are chosen based on the highest customer availabilities. For $c^{\text{failed}} = 1 \cdot \overline{C^{\text{VRP}}}$, the results show that the VRPMTW approaches with the most likely TWs need considerably more vehicles compared to the VRPAP. Only the VRP solution uses the same number of vehicles as the VRPAP solution. The total costs of the VRP solution can be reduced by about 6% when solved as VRPAP. In this case, a slight increase of 2% of the transportation costs C^{trans} (from 710.41 to 725.32) is compensated by a reduction of failed delivery costs C^{failed} of 23% (from 347.26 to 267.00). This suggests that introducing a single or few multiple hard TWs in parcel delivery may only be worthwhile if the costs of failed deliveries are very high (e.g., for perishable goods). To confirm this conclusion, we reproduce the same experiment with more emphasis on failed delivery costs, i.e., $c^{\text{failed}} = 3 \cdot \overline{C^{\text{VRP}}}$ in the bottom section of Table 2.10. In terms of total costs C^{total} , the

Table 2.10: VRPAP vs. VRPMTW

| c^{failed} | Approach | K | C^{trans} | C^{failed} | C^{total} |
|-------------------------------------|---|-------|--------------------|---------------------|--------------------|
| $1 \cdot \overline{C^{\text{VRP}}}$ | VRPAP | 4.84 | 725.32 | 267 | 992.32 |
| | VRPMTW (1 most likely TW) $\hat{=}$ VRPTW | 27.38 | 2009.39 | 85.89 | 2095.28 |
| | VRPMTW (3 most likely TWs) | 10 | 1045.58 | 129.59 | 1175.17 |
| | VRPMTW (5 most likely TWs) | 7.06 | 853.4 | 196.44 | 1049.84 |
| | VRPMTW (10 most likely TWs) $\hat{=}$ VRP | 4.84 | 710.41 | 347.26 | 1057.67 |
| $3 \cdot \overline{C^{\text{VRP}}}$ | VRPAP ($K = K^{\text{VRP}}$) | 4.84 | 763.96 | 733.46 | 1497.42 |
| | VRPAP ($K \leq K^{\text{VRPMTW } 5\text{TWs}}$) | 6.78 | 819.06 | 501.81 | 1320.88 |
| | VRPMTW (1 most likely TW) $\hat{=}$ VRPTW | 27.38 | 2009.39 | 257.68 | 2267.07 |
| | VRPMTW (3 most likely TWs) | 10 | 1045.58 | 388.78 | 1434.36 |
| | VRPMTW (5 most likely TWs) | 7.06 | 853.4 | 589.33 | 1442.73 |
| | VRPMTW (10 most likely TWs) $\hat{=}$ VRP | 4.84 | 710.41 | 1041.79 | 1752.2 |

results look more favorable for the VRPMTW approach with the 3 or 5 most likely TWs, but at the cost of significantly more vehicles needed. When allowing the VRPAP to use the same number of vehicles as in the VRPMTW scenario with 5 TWs, the total costs are considerably lower and the VRPAP does not even use the maximum number of vehicles allowed ($K = 6.78$ against $K = 7.06$).

To further investigate the influence of the number of vehicles on the cost savings potential, we compare the VRPAP with the VRP approach as it is the only one to achieve the same number of vehicles. Figure 2.2 shows three different vehicle scenarios: (0) $K^{\text{VRPAP}} = K^{\text{VRP}}$, i.e., the VRPAP limits the number of available vehicles to the minimum number of vehicles required in the VRP cases, (1) one additional vehicle is available, and (2) two additional vehicles are available. Naturally, the VRPAP modeling approach becomes more important than the VRP approach when


 Figure 2.2: Total costs depending on c^{failed} and the number of vehicles used

the cost of failed-delivery increases. The cost differences are low assuming moderate failed-delivery costs ($c^{\text{failed}} \leq \overline{C^{\text{VRP}}}$, i.e., $x \in [0, 1]$), but become significant if a failed delivery becomes more costly than the average transportation costs per customer in a VRP solution ($c^{\text{failed}} > \overline{C^{\text{VRP}}}$, i.e., $x \in [2, \dots, 10]$). Considering the case with an identical number of vehicles ($K^{\text{VRPAP}} = K^{\text{VRP}}$,

solid line), then at a certain point an increasing failed-delivery cost rate induces limited effects on the solution structure. At this point it becomes much harder to reach TWs with higher customer availability without inducing infeasibility. Thus, if CEPs want to increase their delivery fulfilment even more they have to increase the number of vehicles used (see dashed and dotted lines). The previous experiments have also shown, that in most cases it is not possible to reach all TWs with high or even highest customer availability without increasing the number of vehicles used. Only in an extreme case (i.e., with unlimited vehicles and high c^{failed}) the results achieved by the VRPAP modeling approach become equivalent to those achieved by an approach that first minimizes failed-delivery rates and then solves the resulting VRPTW.

2.6.3.2 Analysis of Availability Profiles

APs may be diverse in reality. It is therefore of interest how the different APs influence the potential cost savings achievable when these are explicitly considered in delivery planning. Figure 2.3 shows the cost savings and the cost distributions of the HALNS results from Table 2.8 compared to the respective upper bounds depending on the type of APs assumed. The lowest cost savings are possible for the A-profiles and the highest for the combination of W- and M-profiles. The share of distance costs increases when the APs are more diverse (AV, AVWM, WM) because in these cases it is worthwhile accepting additional distance in order to reach TWs with higher customer availability, which reduces expected costs for failed delivery. Interestingly, the V-profile also considerably saves costs but does not increase the share of distance costs significantly. This is a promising result since it can be assumed that many customers in reality show a V-profile.

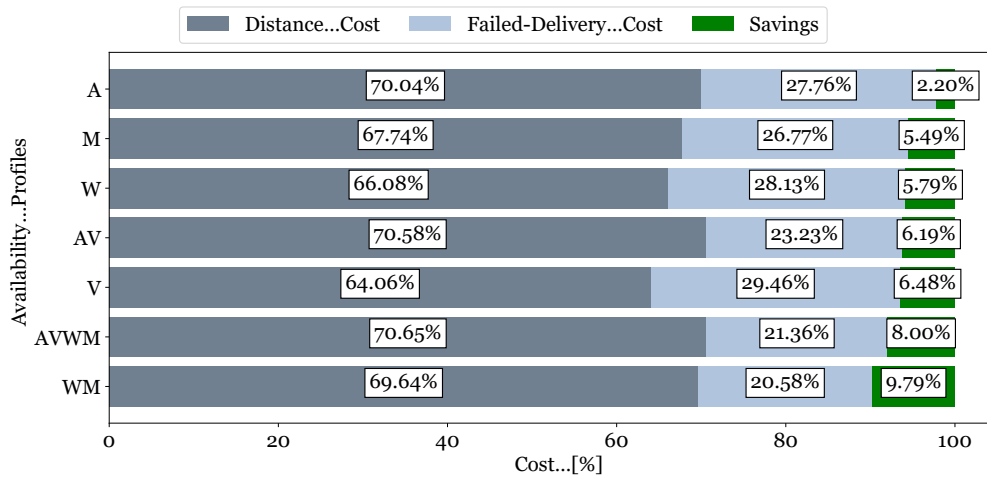


Figure 2.3: Cost savings and cost distributions in delivery planning depending on APs

2.6.3.3 Analysis of Solution Structure

In practice, drivers may change predefined routes on their own initiative if routes seem unreasonable to them. The question therefore arises when tours become apparently too long and drivers have to be motivated to follow the routes suggested even if these look unreasonable at first glance. To obtain an answer to this question, we analyze how the level of the failed-delivery cost parameter

c^{failed} affects the solution structure (see Figure 2.4). Equivalent to the above (see Section 2.6.3.1),

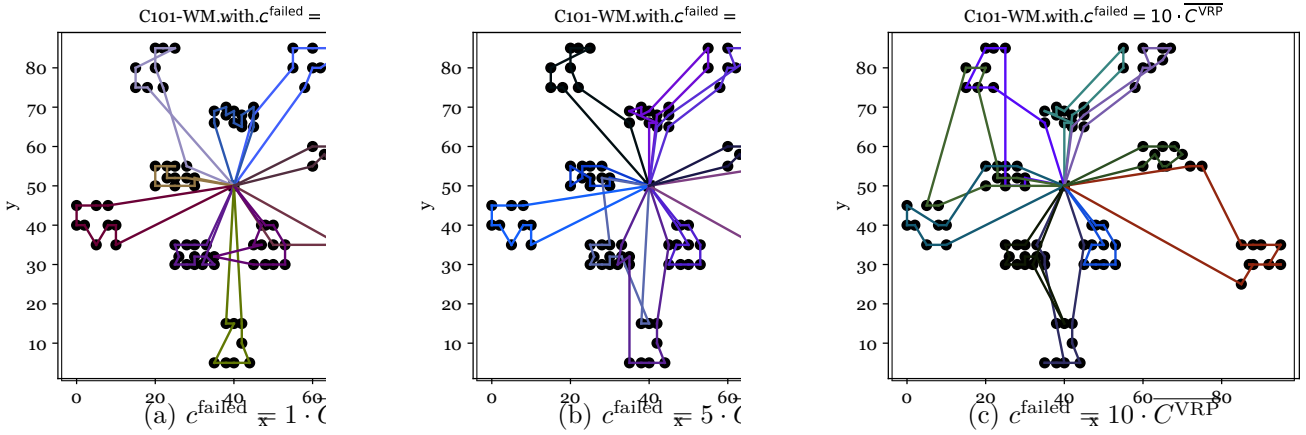


Figure 2.4: Solution structure depending on failed-delivery cost parameter c^{failed} for C101-WM

The routes appear quite reasonable if c^{failed} is low (see Figure 2.4a). The routes only show few crossings and customer clusters are mostly served by the same vehicle. The tours however become less intuitive as c^{failed} increases (see Figures 2.4b and 2.4c). In these cases it is more important to reach a TW with higher customer availability than to save transportation costs. This also affects how customers are combined to tours. A single customer cluster is then even served by several vehicles.

2.6.3.4 Distance Costs vs. Failed-Delivery Rate

In the following we analyze the tradeoff between distance costs and failed-delivery rate. CEPs may be interested in the question regarding to what extent failed-delivery rates can be reduced in exchange for accepting higher transportation costs. To answer this question, we use all instances from Table 2.8. As in Section 2.6.3.1 we apply increasing failed-delivery cost rates ($c^{\text{failed}} = 0 \cdot \overline{C^{\text{VRP}}}, 1 \cdot \overline{C^{\text{VRP}}}, \dots, 10 \cdot \overline{C^{\text{VRP}}}$). In addition, we limit the number of available vehicles to the minimum number of vehicles required in the corresponding VRP cases ($K^{\text{VRPAP}} = K^{\text{VRP}}$).

The lowest possible distance costs are achieved when we neglect failed-delivery costs, i.e., $c^{\text{failed}} = 0$. The average failed-delivery rate then results in a figure above 48%. This case equals the VRP modeling assumptions. Increasing the failed-delivery cost rate c^{failed} to $1 \cdot \overline{C^{\text{VRP}}}$, distance costs rise by about 3%, while the failed-delivery rate is brought down to 38%. The failed-delivery rate can be reduced to a minimum of approx. 34% if we put more emphasis on avoiding failed deliveries by increasing c^{failed} . Distance costs increase simultaneously. The failed-delivery rate improvement however becomes marginal, as it becomes increasingly difficult to reach TWs with higher customer availability without invoking infeasibility. More vehicles would be necessary, as mentioned in Section 2.6.3.1, to further improve the failed-delivery rate.

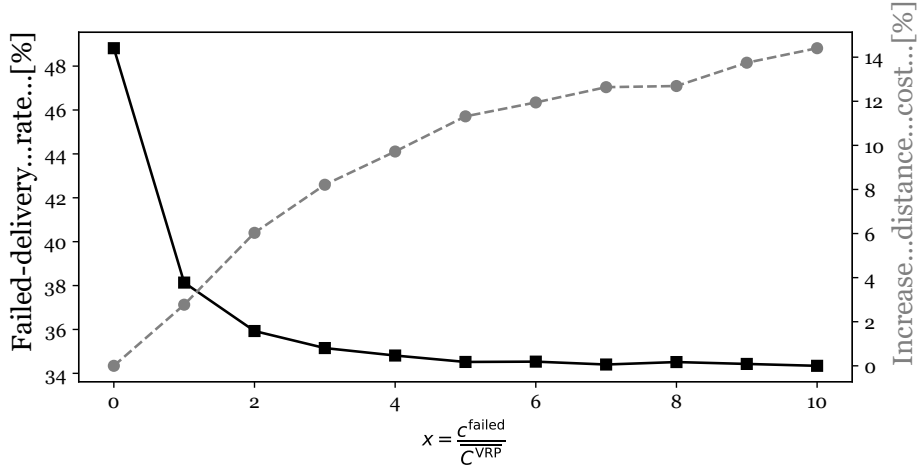


Figure 2.5: Failed-delivery rate vs. increase distance cost

2.6.3.5 Effect of Policies for Failed Deliveries

In this section, we analyze the effect on the subsequent day when the CEP applies the policy to deliver to a neighbor (if available) in the case of a first delivery attempt failing. We randomly decide for each customer if a neighbor is available with 50% probability. We assume that delivery to the neighbor is always successful and therefore no subsequent delivery attempt takes place after the first has failed. To reflect this we assume lower failed-delivery costs of $c^{\text{failed}} = 1 \cdot \overline{C^{\text{VRP}}}$ for customers with a neighbor compared to customers without a neighbor, where we assume $c^{\text{failed}} = 3 \cdot \overline{C^{\text{VRP}}}$ in a first, and $c^{\text{failed}} = 10 \cdot \overline{C^{\text{VRP}}}$ in a second experiment. This results in a ratio of failed-delivery costs for customers with and without a neighbor of either 1:3 or 1:10. We adapt the 42 instances from Table 2.8 and solve them either as VRP or VRPAP with the same maximum number of vehicles. Table 2.11 shows the average results. Columns C^{trans} , C^{failed} and C^{total} represent the transportation, failed and total costs. Column *Failed [%]* shows the failed-delivery rate, i.e., the share of first delivery attempts failing before the neighbor backup option is considered. Column *Neighbor [%]* shows the share of all parcels delivered to a neighbor and Column *Remaining [%]* the deliveries that remain for the delivery on the next day because no neighbor was available. The rows Δ [%] calculate the change in costs or shares, respectively, when comparing the VRP solution with the VRPAP solution.

Table 2.11: Results on VRPAP instances with neighbor delivery for failed deliveries

| | C^{trans} | C^{failed} | C^{total} | Failed [%] | Neighbor [%] | Remaining [%] |
|----------------------|--------------------|---------------------|--------------------|------------|--------------|---------------|
| <i>Neighbor 1:3</i> | | | | | | |
| VRP | 762.87 | 765.73 | 1528.61 | 49.97 | 24.82 | 25.15 |
| VRPAP | 808.90 | 511.16 | 1320.06 | 36.91 | 21.82 | 15.09 |
| Δ [%] | 6.03 | -33.25 | -13.64 | -26.14 | -12.09 | -40.00 |
| <i>Neighbor 1:10</i> | | | | | | |
| VRP | 762.87 | 2111.64 | 2874.51 | 49.97 | 24.82 | 25.15 |
| VRPAP | 870.51 | 1132.18 | 2002.69 | 37.27 | 25.12 | 12.15 |
| Δ [%] | 14.11 | -46.38 | -30.33 | -25.42 | 1.21 | -51.69 |

While the transportation costs C^{trans} increase by around 6% (14%), the failed-delivery costs C^{failed} are brought down by 33% (46%), which leads to a reduction of total costs by around 13% (30%) for a failed-delivery cost ratio of 1:3 (1:10) when using the VRPAP approach. Obviously, the reduction of total costs becomes higher if the cost for a failed-delivery attempt becomes higher. The share of failing deliveries is reduced by around 25%, independent of the failed-delivery cost ratio. While the VRP approach only makes arbitrary use of the neighbor option (for about 50% of failed deliveries), the VRPAP successfully identifies which customers have a neighbor and which do not. In the case of a cost ratio of 1:10, it allows more deliveries to be handed over to a neighbor in order to reach good TWs at customers without a neighbor and achieves more than 50% fewer remaining deliveries, compared to 40% fewer with the cost ratio 1:3. Only 12.15% compared to 15.09% of deliveries remain for the next day. However at higher cost ratios, customers who allow the delivery to a neighbor are increasingly discriminated against. This can be seen in the increasing number of deliveries taken to a neighbor (21% to 25%). This discrimination could dissatisfy these customers in the long run. To prevent this, c_j^{failed} for the customers whose previous delivery has been taken to a neighbor should be adapted for the following periods.

This experiment demonstrates that CEPs benefit in the long run (i.e., for subsequent days) if they use the VRPAP because remaining deliveries can be reduced significantly. Additionally, with the help of the VRPAP and appropriately set c_j^{failed} , CEPs can benefit even more from backup policies. Also, if the maximum number of failed parcels that could be delivered a second time on the next day due to capacity restrictions is known, the VRPAP can be used to respect this limit by setting c_j^{failed} accordingly on each day. We found similar results when considering a parcel shop as backup option instead of a neighbor where failed deliveries are taken to the parcel shop if available.

2.6.4 Case Study with Empirical Availability Profiles

We use the United Kingdom Time Use Survey (UKTUS) 2014-2015 (Gershuny and Sullivan (2017)) to generate realistic APs. The UKTUS is a large-scale survey that provides data on how people in the UK spend their time. Participants of the survey usually record events for a single day. Over a period of 24 hours on a weekday and a day at the weekend, participants indicate their main activity every 10 minutes and specify secondary activities as well as the place where the activity took place and with whom the respondent was together. The location and activity details are used to derive the daily routines of the residents and to identify whether a participant is at home. UKTUS data also makes it possible to discover correlations between this behavior and socio-economic characteristics that are representative for a large population. The UKTUS data set contains information on a total of 4,733 households. This corresponds to 11,422 individuals. We only consider weekdays from Monday to Friday and focus on times from 8-18 o'clock as these represent the main delivery days and periods of parcel service providers. In order to investigate attendance behavior, all persons are removed from the data record who have not provided information on location and activity and are therefore not useful for the creation of APs. The remaining data set contains 7,986 persons aged between 8 and 99 years. It should be noted here that some households may no longer be fully represented with all persons surveyed. In the following, we primarily consider people older than 15,

since it is assumed that younger children generally do not stay at home alone without older siblings or parents or are not allowed to accept parcels. A random sample of 7,221 persons remains. We distinguish between employed (employees and self-employed) and unemployed (including parental leave and the long-term sick) and different age classes. We choose these features as they seem to have the most effect on APs. Appendix A.4 shows ten different APs depending on age and employment status that we derived from UKTUS data and used to generate APs for Solomon instances (R1, R2, C1, C2, RC1 and RC2). We randomly sample from these APs and allocate each customer one of these ten APs in the Solomon instances. The probability of assigning a specific AP to a customer depends on the relative frequency of this profile, e.g., the relative frequency is calculated by $\frac{675}{7221} = 0.0935$ for employed people between 15 and below 30. The sampling is repeated 30 times, resulting in 30 AP distributions for each of the six geographical distributions of customers from Solomon, leading to 180 instances overall.

We solve all instances ten times with HALNS with $c^{\text{failed}} = 3 \cdot \overline{C^{\text{VRP}}}$ and present the average (over 30 instances per geographical distribution) of the best result encountered during these 10 runs in Table 2.12. On average the distance cost increases from 762.88 to 805.45 whereas the rate of failed deliveries decreases from 0.585 to 0.5156. This corresponds to 5.58% rise in distance costs in order to decrease the failed-delivery rate by 11.86%. Total expected costs decrease by 4.82% from 2103.56 to 2002.15. A Wilcoxon test confirmed that all differences in total costs and in the share of failed deliveries are highly significant ($p < 0.001$). To summarize, costs are reduced by about 5% by only using information on the age and employment status. This confirms the value of the VRPAP even without using sophisticated methods to determine customer individual APs.

Table 2.12: Average Results on Solomon Instances with UK TUS APs

| Instance | $C_{\text{VRP}}^{\text{Dist}}$ | $P_{\text{VRP}}^{\text{Failed}}$ | $C_{\text{VRP}}^{\text{Failed}}$ | $C_{\text{VRP}}^{\text{Total}}$ | $C_{\text{VRPAP}}^{\text{Dist}}$ | $P_{\text{VRPAP}}^{\text{Failed}}$ | $C_{\text{VRPAP}}^{\text{Failed}}$ | $C_{\text{VRPAP}}^{\text{Total}}$ |
|-----------|--------------------------------|----------------------------------|----------------------------------|---------------------------------|----------------------------------|------------------------------------|------------------------------------|-----------------------------------|
| R101-TUS | 865.95 | 0.5873 | 1525.72 | 2391.67 | 872.49 | 0.5597 | 1453.92 | 2326.41 |
| R201-TUS | 651.3 | 0.5873 | 1147.53 | 1798.83 | 725.03 | 0.4519 | 882.89 | 1607.92 |
| C101-TUS | 824.78 | 0.5796 | 1434.13 | 2258.91 | 830.49 | 0.5384 | 1332.24 | 2162.73 |
| C201-TUS | 584.28 | 0.5745 | 1007.01 | 1591.29 | 667.11 | 0.5049 | 885.08 | 1552.19 |
| RC101-TUS | 995.59 | 0.5949 | 1776.83 | 2772.42 | 1008.37 | 0.5719 | 1708.04 | 2716.41 |
| RC201-TUS | 655.35 | 0.5864 | 1152.89 | 1808.24 | 729.19 | 0.467 | 918.06 | 1647.24 |
| Average | 762.88 | 0.585 | 1340.69 | 2103.56 | 805.45 | 0.5156 | 1196.71 | 2002.15 |

2.7 Conclusions and Future Areas of Research

Conclusions In this paper we present a time-constrained vehicle routing problem, the vehicle routing problem with availability profiles that addresses the tradeoff between routing and expected failed-delivery costs. We use so-called availability profiles that indicate the probability of successful delivery for a set of potential delivery TWs. We show how these APs can be integrated into an optimization model for delivery route planning such that routing and expected failed-delivery costs are considered simultaneously. We propose a novel hybrid adaptive large neighborhood search solution framework for the VRPAP and related time-constrained VRPs. The HALNS embeds an

ALNS into a population-based metaheuristic. The HALNS shows results on a par with the best-performing algorithms from the literature. For the VRPMTW it finds 12 and for the VRPSTW 8 new BKS. The hybrid approach shows superiority to sole ALNS implementations. We also undertake diverse numerical studies on newly generated VRPAP instances based on simulated and real-world data that confirm the value of integrated planning of delivery tours and delivery TWs. These studies additionally reveal several managerial insights:

- (a) The VRPAP improves the total cost by 6% on average compared to the best-performing benchmark approaches that are potentially applied in practice. The number of vehicles may have to be increased to exhaust all cost benefits when tours and TWs are planned simultaneously.
- (b) More peaks within APs lead to higher cost savings. An A-profile thus offers fewer opportunities for cost improvement than W- and M-profiles or a combination of these. V-profiles that can frequently be observed in reality considerably save total costs without increasing distance costs significantly.
- (c) In cases where delivery fulfilment has a high priority, overall transportation costs can substantially increase. Customers in the same region could even be served by several vehicles in such instances. Drivers must then be motivated to follow their routes, even if the tours created seem unreasonable at first glance.
- (d) A small increase in transportation costs may noticeably increase delivery fulfilment. Above a certain level, however, a further increase in delivery fulfilment leads to disproportionately high additional transportation costs.
- (e) CEPs may benefit from using the VRPAP on subsequent days because the number of remaining deliveries can be reduced significantly, especially if there are additional backup options available.
- (f) Generally available socio-economic data are already sufficient to define APs of customers. In an experiment close to reality the failed-delivery rate is reduced by about 12% while overall costs decrease by about 5%.

Outlook on Future Research There are several opportunities for improvement and extension based on the proposed modeling and solution approach. The determination of APs could be further developed, e.g., by focusing more intensively on individual customer behaviors. Also, customer presence is becoming a critical success factor for new delivery technologies, such as drone parcel delivery or mobile parcel lockers (see, e.g., Schwerdfeger and Boysen 2020). The VRPAP modeling and solution approach could be extended in these directions. Finally, as the HALNS framework mostly uses simple operators, it could be easily adapted to solve related routing problems, while the benefits from hybridization would be retained.

**Contribution 3 - Hybrid Adaptive Large
Neighborhood Search for Vehicle Routing
Problems with Depot Location Decisions**

3 Hybrid Adaptive Large Neighborhood Search for Vehicle Routing Problems with Depot Location Decisions

Stefan Voigt, Markus Frank, Pirmin Fontaine, Heinrich Kuhn

Abstract This article considers three variants of the vehicle routing problem (VRP). These variants determine the respective depot locations from which customers are supplied, i.e., the two-echelon VRP (2E-VRP), the location routing problem (LRP), and the multi-depot VRP (MDVRP). Both the LRP and the MDVRP can be formulated as special cases of the 2E-VRP, so that all three problem classes can be readily solved via a single solution approach. We develop such a unified solution approach for all three problem classes based on the recently proposed hybrid adaptive large neighborhood search (HALNS). The HALNS uses a population of solutions generated by an efficient ALNS. Individuals of this population are subject to a crossover and selection phase, using elements of genetic algorithms resulting in a hybrid heuristic. Computational experiments on several sets of instances from literature demonstrate the competitive performance of the HALNS. The HALNS outperforms all approaches for solving the 2E-VRP and is on par with heuristics that are dedicated either to the LRP or the MDVRP. Furthermore, the HALNS shows superior robustness, i.e., the variance of results from several runs is comparatively low. The HALNS especially outperforms all existing pure ALNS implementations on these problem classes, demonstrating the value of hybridization. Additionally, the HALNS finds three new best-known solutions for LRP instances.

Accepted in *Computers & Operations Research*, 105856

URL: <https://doi.org/10.1016/j.cor.2022.105856>

3.1 Introduction

The multi-depot vehicle routing problem (MDVRP) can be formulated as a special case of the location routing problem (LRP), and the LRP in turn as a two-echelon vehicle routing problem (2E-VRP) (Hemmelmayr et al. 2012). We classify these problems as *Vehicle Routing Problems with Depot Location Decisions*. Besides the clustering and sequencing as essential decisions in any vehicle routing problem (VRP), problem settings in this class are characterized by decisions about which of multiple depots to open and which customers to serve from each depot opened.

The most generic problem in this class is the 2E-VRP. The 2E-VRP arises for example in city logistics where goods are transported from a main depot outside the city to several satellites within the city, from where the goods are finally distributed to customers (e.g., Crainic et al. 2011). Two interdependent decisions have to be made in order to solve the 2E-VRP. First, we have to cluster customers to satellites and decide about the routing from the main depot to satellites (first level). Not all potential satellites necessarily have to be used. Second, we need to decide about the routing from satellites to customers (second level). We call the first decision *Depot Location Decision*, which includes the first-level routing, and the second decision *Routing Decision*. The MDVRP and LRP share these decisions with the 2E-VRP and can be interpreted as special cases of the 2E-VRP. The *Routing Decision* covers a classical VRP that is already \mathcal{NP} -hard (e.g., Vidal et al. 2013a). The 2E-VRP, LRP, and MDVRP themselves are therefore \mathcal{NP} -hard.

This paper, develops a unified heuristic solution approach to solve all three problems. Based on recent work by Voigt et al. (2021), we adapt and enhance the concept of a hybrid adaptive large neighborhood search (HALNS) that uses multiple solutions generated by an efficient ALNS as a population of individuals within a genetic algorithm framework. Numerical experiments based on established benchmark data sets for the 2E-VRP, LRP, and MDVRP show the robustness and good performance of the HALNS.

Our research contributes to the existing literature as follows. (1) We propose a novel solution approach that hybridizes an ALNS with elements of genetic algorithms for solving these problems. (2) We present a performance ranking of all heuristics reviewed from the literature for the problems named and compare their solution quality and scaled computation times. (3) We show that the HALNS outperforms approaches for the 2E-VRP and is on par with problem-specifically designed solution approaches for the LRP and MDVRP. Furthermore, the HALNS outperforms sole implementations of state-of-the-art ALNSs, making hybridization an interesting option to improve existing ALNS implementations. (4) We present insights on the components of the HALNS, showing the importance of using these components.

The remainder of this paper is structured as follows: Section 3.2 describes the vehicle routing problems with depot location decisions that we address with our heuristic and their relationships. In Section 3.3, we cluster previous heuristic solution approaches for these problems and analyze the frequency of heuristic components. We present the HALNS developed in Section 3.4 and conduct computational experiments in Section 3.5, including the evaluation of previous approaches and their comparison with the HALNS based on established benchmark sets. Finally, we conclude our work and indicate further directions for research in Section 3.6.

3.2 Vehicle Routing with Depot Location Decisions

We consider three VRP classes that determine the depot locations, namely the 2E-VRP, as well as the LRP and MDVRP as special cases of the 2E-VRP. These problems have in common the decision from which location, i.e., satellite, facility or depot, customers are served. We will henceforth use the term depot except when referring to the specific problem class, i.e., we use satellite when referring to 2E-VRP, facility for LRP, and depot for MDVRP. We first explain the 2E-VRP in detail in Section 3.2.1. Afterwards, we show how the LRP and MDVRP can be interpreted as special cases of the 2E-VRP in Section 3.2.2.

3.2.1 2-Echelon VRP

The 2E-VRP is a two-level routing problem with two fleets of vehicles, which minimizes costs arising from servicing customers via given satellites from a single main depot. First-level vehicles start from the main depot to satellite locations. Freight is then transferred from first-level vehicles at the satellite to second-level vehicles. The first-level routing problem constitutes a VRP with split deliveries, as it is possible that more than one vehicle delivers to a single satellite because of capacity restrictions. Second-level vehicles start from satellites to customers and execute the final delivery. Note that split deliveries are not allowed on the second level. The second level therefore represents a MDVRP, which is explained in more detail in Section 3.2.2.

The 2E-VRP is defined on a directed graph $G = (V, A)$. V is the set of nodes, consisting of the main depot subset $V_0, V_0 = \{v_0\}$, the satellite subset V_S with n_s satellites s , and the customer subset V_C with n_c customers i . Each customer node has an associated demand d_i . The arc set A consists of one set $A_1 = \{(i, j) : i, j \in V_0, V_S\}$ with arcs connecting the depot node with satellite nodes and the satellite nodes amongst each other, and a second set $A_2 = \{(i, j) : i, j \in V_S, V_C\}$ with arcs connecting satellite nodes with customer nodes and customer nodes amongst each other. This means that direct deliveries from the depot to customers are not possible. We assume a positive weight of arc (i, j) , c_{ij} , that reflects the distance, traveling times or transportation costs between nodes i and j . Vehicles are limited in their number and have limited capacity on both levels. m_1 identical vehicles are available at the main depot for the first level and have a capacity of K_1 each. m_2 identical vehicles with capacity K_2 each are available in total for the second level. The satellites are implicitly capacitated by constraining the number of second-level vehicles m_s allowed to start from the satellite s , or explicitly by a capacity limit w_s . The flow of demand to and from each satellite must be balanced. Figure 3.1 illustrates an example for a 2E-VRP instance, with customers receiving their deliveries via satellites s_1, s_2, s_3 out of four possible satellites, which are in turn supplied by depot v_0 , applying weights c_{ij} on the first level.

The objective in the 2E-VRP is to minimize the routing costs arising from the vehicle fleets traveling on the first and the second level by 1) assigning customers to satellites, 2) deciding on the clustering of customers to tours and their sequence on each tour for each satellite, and 3) the routing from the main depot to satellites, respecting the flow of demand. Perboli et al. (2011) present a corresponding MIP model.

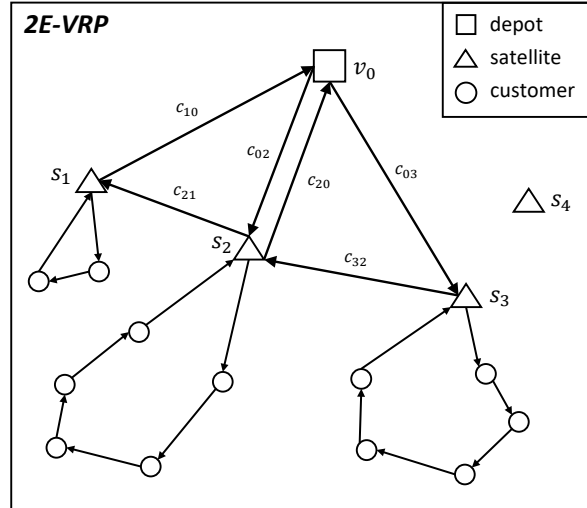


Figure 3.1: Exemplary 2E-VRP instance

3.2.2 Location Routing Problem and Multi-Depot VRP as Special Cases of the 2E-VRP

Location Routing Problem The LRP can be described as a strategic problem that consists of the facility location problem and the (multi-depot) vehicle routing problem. The LRP aims for the cost optimal locations of multiple facilities while considering the resulting routing of vehicles. Obviously these decisions interact with each other. In the following we consider the standard LRP, defined as a deterministic, static, discrete, single-echelon problem that minimizes costs arising from fixed costs f_s for opening (capacitated) facilities and travel costs in order to serve customers exactly once via one vehicle (Schneider and Drexl 2017). Figure 3.2 shows how an LRP instance can be formulated as a special case of the 2E-VRP presented above. The following adaptations must be made (see, Hemmelmayr et al. 2012):

- Facilities in the LRP correspond to satellites V_s in the 2E-VRP.
- A dummy main depot v_0 is created. Edges from the main depot to facilities V_s are respectively valued with fixed costs $c_{0s} = f_s$ for opening the facility $s \in V_S$. Edges from one facility to another are valued with a sufficiently large number. There is no cost on the return trip to the main depot v_0 , i.e., $c_{s0} = 0$. A facility is thus opened when a first level vehicle visits that facility.
- m_1 is set to the number of facilities available.
- K_1 is set to the maximum capacity of all facilities to choose from. This ensures that split deliveries are not necessary and every facility can be served by exactly one vehicle, so the opening costs are only charged once.

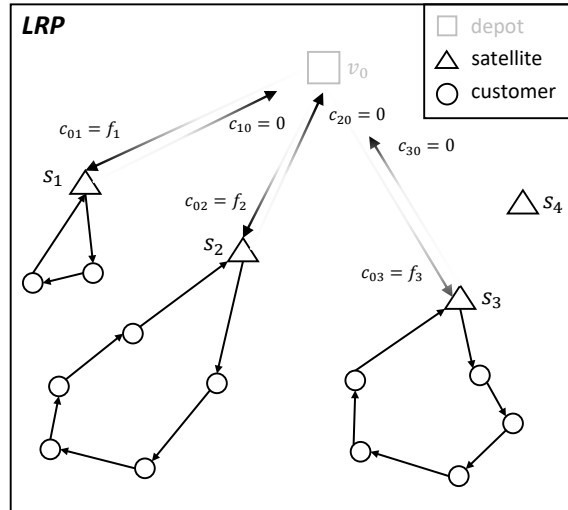


Figure 3.2: Exemplary LRP instance as special case of the 2E-VRP

Multi-Depot VRP In the MDVRP each customer is served exactly once from one of several available depots. Vehicles are assigned to a depot and must therefore start and end at the same depot. Depots do not have opening costs. Renaud et al. (1996), for example, present a mathematical model of the MDVRP. The MDVRP can be formulated as 2E-VRP by introducing a dummy main depot and treating all other depots as satellites. All travel costs on the first level are set to zero, i.e., $c_{s,0} = c_{0,s} = c_{st} = 0 \forall s, t \in V_s$, see also Figure 3.3. As mentioned in Section 3.2.1, the second level of the 2E-VRP is an MDVRP, making it obvious that the MDVRP is a special case of the 2E-VRP without considering the first-level routing.

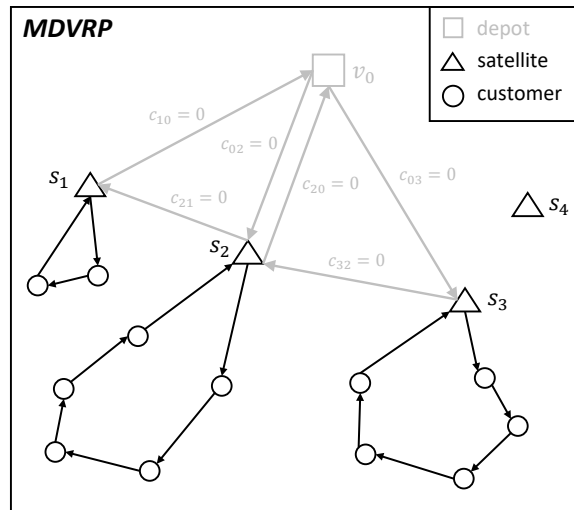


Figure 3.3: Exemplary MDVRP instance as special case of the 2E-VRP

3.3 Overview of Related Heuristics

All three problem classes considered in this study, the 2E-VRP, LRP and MDVRP, are known to be \mathcal{NP} -hard problems. The literature therefore suggests a diverse set of heuristic solution approaches. In this section we categorize and discuss the available approaches, focusing on those that have been applied to at least one of the commonly known benchmark instances (see Section 3.5.1). The review of solution approaches for each problem class is mainly structured according to the dominant solution approach, i.e., matheuristic or metaheuristic. Furthermore, we distinguish metaheuristic approaches with respect to their search space, i.e., neighborhood-based, population-based, or hybrid search space. In the following, we briefly clarify our understanding of these distinctions.

- **Heuristic approach**

- *Matheuristic*: Combines exact and (meta)heuristic components.
- *Metaheuristic*: Uses strategies to guide other heuristic components.

- **Search space of metaheuristics**

- *Neighborhood-based*: Focuses on one incumbent solution.
- *Population-based*: Uses a population of several solutions, so-called individuals.
- *Hybrid*: Traverses the search space by using neighborhood-based and population-based means. Please note that merely combining local search with other approaches does not qualify for our classification as hybrid.

We focus on heuristic approaches for the standard versions of the respective problems. Exact approaches are out of scope for this review. We refer to the contributions of Marques et al. (2020) for the 2E-VRP, to Contardo et al. (2014) for the LRP and to Sadykov et al. (2021) for the MDVRP. For additional in-depth investigations and complete literature reviews we refer to Sluijk et al. (2022) for the 2E-VRP, to Schneider and Drexl (2017) for the LRP, to Drexl and Schneider (2015) and Prodhon and Prins (2014) for LRP variants and to Montoya-Torres et al. (2015) for the MDVRP.

3.3.1 2-Echelon VRP

Matheuristics Perboli et al. (2011) introduce a mathematical model for the 2E-VRP and strengthen their formulation with valid inequalities. They also develop two matheuristics based on the mathematical model and the observation that the 2E-VRP can be split into $n_k + 1$ VRPs, which in turn can be solved by any exact method or heuristic for the VRP. The authors therefore focus on finding near-optimal assignments for customers to satellites. Wang et al. (2017) propose a variable neighborhood search (VNS) and a post-optimization step based on integer programming to solve the 2E-VRP with environmental considerations. Amarouche et al. (2018) combine two components to solve the classical 2E-VRP. The first component generates a set of routes by a large neighborhood search (LNS). The second component uses integer programming to recombine the routes. Jie et al. (2019) decompose the problem by solving the first echelon with an exact column generation approach and the computationally more demanding second echelon with an ALNS largely based on Hemmelmayr et al. (2012) and Ropke and Pisinger (2006).

Neighborhood-based Metaheuristics Neighborhood-based approaches for the 2E-VRP usually tackle the first- and second-level routing separately. Crainic et al. (2011) present a multi-start heuristic based on this two-level approach. The approach iteratively solves the two subproblems by applying a local search, perturbing the solution and applying a feasibility search, if necessary. Hemmelmayr et al. (2012) develop an ALNS with specially designed operators for the 2E-VRP and the LRP. The heuristic works in a sequential manner like the approach presented previously. The authors distinguish between operators with large and small impact on the solution structure. Operators with a large impact open or close satellites, while operators with a small impact only execute minor changes. Solutions destroyed by operators with a large impact are accepted as a new incumbent solution, even if the solution is worse. In contrast, solutions destroyed by operators with a small impact must be better in order to be accepted. Hence, the solution space is explored by the large-impact operators, and the search intensifies by applying small-impact operators. Enthoven et al. (2020) and Yu et al. (2021) present similar ALNSs for variants of the 2E-VRP with a large set of small- and large impact operators. Compared to the rather sophisticated ALNSs, Breunig et al. (2016) implement a simplistic LNS with only few operators. They also provide consistent benchmark instances, resolving some confusion with previously existing instances.

Hybrid Metaheuristics The parallelized LNS (PLNS) of Mühlbauer and Fontaine (2021) is currently the only approach for the 2E-VRP hybridizing an LNS with a population of solutions. The authors develop the PLNS for solving a problem arising in city logistics when using cargo bicycles with swap containers, which can be modeled as a 2E-VRP. The authors suggest a new first-level heuristics and show the value of incorporating first-level costs when making decisions at the second level.

3.3.2 Location Routing Problem

A large variety of heuristics for the LRP exist. Many of these approaches rely on a sophisticated combination of solution approaches; only few approaches like the tabu search of Tuzun and Burke (1999) or the simulated annealing (SA) of Yu et al. (2010) are straightforward implementations. Combinations of approaches are often based on a greedy randomized adaptive search procedure (GRASP) (Prins et al. (2006b), Duhamel et al. (2010), Contardo et al. (2013)) or a granular tabu search (GTS) (Prins et al. (2007), Escobar et al. (2013), Escobar et al. (2014a) and Schneider and Löffler (2019)).

Matheuristics Prins et al. (2007) implement a heuristic based on a GTS and a lagrangean relaxation to solve the LRP. Pirkwieser and Raidl (2010) tackle the LRP and periodic LRP (PLRP) with a VNS combined with three very large neighborhood searches based on integer linear programming (ILP) similar to Prins et al. (2007). Contardo et al. (2013) present a three-phase heuristic that (1) applies a GRASP based on Prins et al. (2006b) to construct initial solutions, (2) solves an ILP in order to recombine routes from the set of initial solutions similar to Pirkwieser and Raidl (2010), and (3) solves the same ILP to improve the solution.

Neighborhood-based Metaheuristics The tabu search of Tuzun and Burke (1999) works in two phases. First the location phase tries to find a good location configuration, and second the routing phase minimizes routing costs. The authors propose a set of benchmark instances that laid the foundation for comparing heuristics for the LRP. Yu et al. (2010) use a simplistic SA metaheuristic based on a special solution representation, encoding the solution in only one string. The problem is then solved by common neighborhood operators in an integrated manner compared to previous publications that tackled the problem by separating the location and routing decisions.

Escobar et al. (2013) propose a two-phase heuristic using a construction phase and as second phase a GTS with different diversification strategies. The heuristic uses a software library containing fast local search heuristics for the VRP, and works especially well on larger instances. A similar team of authors present a granular variable tabu neighborhood search, that combines a GTS with a VNS (Escobar et al. 2014a). Schneider and Löffler (2019) present a tree-based search algorithm. The algorithm uses a location phase that searches for a good set of facilities in a tree-like fashion and a routing phase that solves the resulting MDVRP by a GTS. The GTS is composed of a large set of 14 neighborhood operators. In contrast to that, the approach of Arnold and Sörensen (2021) uses only few operators based on the knowledge-guided local search (Arnold and Sörensen 2019). In addition, a progressive filtering technique significantly reduces the computational effort wasted on calculating routing solutions on unpromising depot configurations by estimating an upper bound of the number of open depots.

Population-based Metaheuristics Population-based approaches are again rarely used for the LRP, although the approaches of Ting and Chen (2013) and Lopes et al. (2016) show good results on benchmark instances. Prodhon and Prins (2008) implement a memetic algorithm, i.e., a genetic algorithm with local search, with population management for the PLRP and for the LRP (Prins et al. 2006a). Ting and Chen (2013) implement a multiple ant colony optimization algorithm (ACO) with two hierarchical ant colonies. The first colony solves the facility location problem, the second colony the MDVRP. Both colonies exchange information through a global pheromone updating rule. Lopes et al. (2016) propose a genetic algorithm with local search procedures as mutation operators and a so-called route copy crossover operator, which always maintains feasibility.

Hybrid Metaheuristics Prins et al. (2006b) propose a GRASP combined with a learning process, and a post-optimization step using path relinking. The authors were to the best of our knowledge the first to test the performance on three sets of instances from Tuzun and Burke (1999), Prins et al. (2004) and Barreto et al. (2007). Duhamel et al. (2010) propose a GRASP that uses an evolutionary local search within two solution spaces.

3.3.3 Multi-Depot VRPs

Matheuristics Subramanian et al. (2013) develop a matheuristic based on a set partitioning formulation and an iterated local search (ILS) for a large class of VRPs, including the MDVRP. The MIP solver and the ILS cooperate while solving the set partitioning problem.

Neighborhood-based Metaheuristics The concept of tabu search has also been frequently applied to the MDVRP. Early approaches are from Renaud et al. (1996) and Cordeau et al. (1997). Renaud et al. (1996) first construct an initial solution similar to the approach of Chao et al. (1993), except that they use a petal heuristic for solving the VRPs instead of the savings algorithm. Second, the initial solution is improved by applying a tabu search. The tabu search of Cordeau et al. (1997) is developed for solving the PVRP. They show that the periodic traveling salesman problem and the MDVRP can be seen as special cases of the PVRP and therefore readily solved with the same tabu search. More recently Escobar et al. (2014b) adapt the hybrid granular tabu search from Escobar et al. (2013) to solve the MDVRP. Cordeau and Maischberger (2012) implement a parallel iterated tabu search for solving the VRP, the PVRP, MDVRP, and the site-dependent VRP. The heuristic combines ILS with tabu search. The authors show that the implementation can be readily implemented in parallel. Sadati et al. (2021) implement a granular tabu search combined with a VNS for solving a class of MDVRPs. The approach uses a shaking phase with a large set of neighborhood structures. As another neighborhood-based approach, Pisinger and Ropke (2007) use the ALNS for solving five different VRP variants, including the MDVRP. All problems are transformed to a pickup and delivery problem and solved with the ALNS presented in Ropke and Pisinger (2006). The ALNS adaptively chooses removal and insertion operators to build new solutions.

In contrast to the rather sophisticated metaheuristic approaches described above, Arnold and Sörensen (2019) propose a knowledge-guided local search with few efficiently pruned local search operators. The local search uses only a simple perturbation of the cost matrix to escape local optima.

Population-based Metaheuristics The ACO of Yu et al. (2011) uses an ant-weight strategy and mutation operators. The approach adds a virtual central depot as the origin to make the MDVRP similar to a classical VRP. Luo and Chen (2014) present a nature-inspired algorithm, which can be classified as a particle swarm optimization algorithm (PSO). The HGSADC from Vidal et al. (2012) is also able to solve the multi-depot periodic VRP, which includes the MDVRP and PVRP as special cases. The metaheuristic uses two populations of individuals, one for feasible and one for infeasible solutions. Individuals are educated, i.e., a local search is applied on offsprings. The diversity of the population is maintained by evaluating the fitness from a cost perspective and a distance measure to every other individual. Vidal et al. (2014a) combine sequence-based moves with an optimal choice of vehicle, depot and of the first customer to be visited in the route. They present a dynamic programming approach for the evaluation of these neighborhoods. The authors integrate this approach into an ILS and into the HGSADC based on Vidal et al. (2012), demonstrating the value of the proposed concept.

3.3.4 Summary

Numerous heuristic approaches can be found in the literature for the interrelated problem settings of 2E-VRP, LRP and MDVRP. Mostly, they are specifically developed for a single problem class without considering the related problem settings. We give an overview of solution approaches

suggested and their respective components applied in Table 3.1. As already mentioned, we classify the approaches as matheuristic or metaheuristic and denote the respective search space used in a metaheuristic, either neighborhood-based, population-based or both, i.e., hybrid search space. In addition, Table 3.1 denotes various heuristic components that are used in the respective solution approaches. The latter classification scheme is similar to Vidal et al. (2013a).

- **Local improvement:** Solutions are improved by local search procedures, e.g., 2-opt.
- **SA:** Simulated annealing accepts deteriorating solutions with a probability controlled by a gradually decreasing temperature parameter.
- **Tabu search:** Some elements/moves are set tabu for a certain period.
- **Granularity:** The arc set is reduced, e.g., a certain percentage of the longest arcs are excluded.
- **Indirect representations:** The solution is not directly represented, but must be extracted from the indirect representation by applying a decoding procedure, e.g., by applying a split procedure on a giant tour representation.
- **Infeasible solutions:** During the search, infeasible solutions may occur (e.g., if the capacity of vehicles is violated) and be accepted as an incumbent solution. Infeasible solutions are usually penalized by a cost factor or kept in a different population.
- **Diversity management:** The diversity of a population is managed by introducing a distance measure between individuals and using this measure for selection purposes.
- **Parameter adaptation:** Parameters are adapted during the search, e.g., weights for operators (ALNS) or the penalty factor in the presence of infeasible solutions.

Table 3.1 shows that hybrid approaches and also approaches that include multiple components were rarely studied. Almost all publications present results and runtimes on established benchmark instances that we detail in Section 3.5.1. This facilitates a detailed comparison of the performance of these approaches in Section 3.5.3, where we determine state-of-the-art heuristics for benchmarking the HALNS. The HALNS uses a unique mix of components that has not been considered before. We evaluate the impact of single HALNS components in Section 3.5.4.

Table 3.1: Classification of heuristics solving 2E-VRP, LRP, or MDVRP

| Approach/Authors | Classification | | | | Components | | | | | | | | | | | | | |
|---------------------------------|----------------|---------------|--------------------|------------------|-------------------|----|-------------|-------------|----------------------|--------------------------|----------------------|----------------------|---|----|----|---|---|----|
| | Class | Metaheuristic | Neighborhood-based | Population-based | Local Improvement | SA | Tabu Search | Granularity | Infeasible solutions | Indirect representations | Diversity management | Parameter adaptation | | | | | | |
| 2E-VRP | | | | | | | | | | | | | | | | | | |
| Yu et al. (2021) | x | x | x | | x | | | | | | | x | | | | | | |
| Mühlbauer and Fontaine (2021) | x | | | | | | | | | | x | | | | | | | |
| Enthoven et al. (2020) | x | x | x | | | | | | | | x | | | | | | | |
| Jie et al. (2019) | | x | x | | | | | | | | | x | | | | | | |
| Amarouche et al. (2018) | | x | x | | | | | | | | | x | | | | | | |
| Wang et al. (2017) | | x | | | | | | | | | | x | | | | | | |
| Breunig et al. (2016) | | x | x | | | | | | | | | x | | | | | | |
| Hemmelmayr et al. (2012) | x | x | x | | | | | | | | | x | | | | | | |
| LRP | | | | | | | | | | | | | | | | | | |
| Arnold and Sörensen (2021) | x | x | x | | | | | | | | | x | | | | | | |
| Schneider and Löffler (2019) | x | x | x | | | | | | | | | x | | | | | | |
| Lopes et al. (2016) | x | x | x | | | | | | | | | x | | | | | | |
| Escobar et al. (2014a) | x | x | x | | | | | | | | | x | | | | | | |
| Ting and Chen (2013) | x | x | x | | | | | | | | | x | | | | | | |
| Escobar et al. (2013) | x | x | x | | | | | | | | | x | | | | | | |
| Contardo et al. (2013) | x | x | x | | | | | | | | | x | | | | | | |
| Hemmelmayr et al. (2012) | x | x | x | | | | | | | | | x | | | | | | |
| Yu et al. (2010) | x | x | x | | | | | | | | | x | | | | | | |
| Dubamel et al. (2010) | x | x | x | | | | | | | | | x | | | | | | |
| Prins et al. (2007) | x | x | x | | | | | | | | | x | | | | | | |
| Prins et al. (2006a) | x | x | x | | | | | | | | | x | | | | | | |
| Prins et al. (2006b) | x | x | x | | | | | | | | | x | | | | | | |
| MDVRP | | | | | | | | | | | | | | | | | | |
| Sadati et al. (2021) | x | x | x | | | | | | | | | x | | | | | | |
| Arnold and Sörensen (2019) | x | x | x | | | | | | | | | x | | | | | | |
| Luo and Chen (2014) | x | x | x | | | | | | | | | x | | | | | | |
| Vidal et al. (2014a) | x | x | x | | | | | | | | | x | | | | | | |
| Escobar et al. (2014b) | x | x | x | | | | | | | | | x | | | | | | |
| Subramanian et al. (2013) | x | x | x | | | | | | | | | x | | | | | | |
| Vidal et al. (2012) | x | x | x | | | | | | | | | x | | | | | | |
| Cordeau and Maischberger (2012) | x | x | x | | | | | | | | | x | | | | | | |
| Yu et al. (2011) | x | x | x | | | | | | | | | x | | | | | | |
| Pisinger and Ropke (2007) | x | x | x | | | | | | | | | x | | | | | | |
| Sum | 6 | 25 | 21 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 29 | 7 | 8 | 12 | 13 | 8 | 7 | 13 |
| This article | x | | | x | | | | | | | | | | | | | | |

3.4 Hybrid Adaptive Large Neighborhood Search for VRPs with Depot Location Decisions

Despite the existence of numerous specialized approaches for solving the 2E-VRP, LRP and MD-VRP, there is no method proven to be capable of solving all three problem classes equally efficiently and effectively. We develop a new solution approach, the hybrid adaptive large neighborhood search (HALNS) based on the solution framework proposed by Voigt et al. (2021) that can be used for all three problem classes. Algorithm 6 depicts the general scheme of the HALNS. The HALNS uses multiple solutions, i.e., individuals, generated by an ALNS, called population P , and executes crossover steps within that population. The initial population consists of n^P individuals, each generated by executing an ALNS run (lines 1-3 of Alg. 6). In further generations, the ALNS is used to crossover and educate individuals (line 6-9 of Alg. 6). Surviving individuals are chosen according to solution quality and contribution to the diversity of the population (line 10 of Alg. 6). Depot locations can be set tabu for the next generation if they are not used within at least one of the individuals or selected by chance (line 11 of Alg. 6). Figure 3.4 additionally illustrates such a single HALNS generation. The procedure ends when gen^{stop} generations without finding a new best solution have been executed (line 4 of Alg. 6) or the maximum number of generations gen^{max} is reached. The following sections explain the components of the HALNS in more detail and show the problem-specific adaptations.

Algorithm 6: Hybrid adaptive large neighborhood search

```

1 while  $|P| < n^P$  do // Initial population (3.4.1)
2    $s \leftarrow \text{ALNS}()$ 
3    $P \leftarrow P \cup \{s\}$ 
4 while  $gens \text{ without Improvement} < gen^{stop} \wedge gens < gen^{max}$  do // GA generations
5    $\hat{s} \leftarrow \text{DetermineBestSolution}(P)$ 
6   while  $i < n^P$  do // Crossover and education phase (3.4.2)
7      $s \leftarrow P[i]$ 
8      $s \leftarrow \text{ALNS}(s, \hat{s})$ 
9      $P \leftarrow P \cup \{s\}$ 
10   $P \leftarrow \text{DiversityManagement}(P)$  // Select survivors and manage diversity (3.4.3)
11   $\text{SetDepotLocationsTabu}(P)$  // Set depot locations tabu for next generation (3.4.4)

```

3.4.1 ALNS Algorithm

Algorithm 7 describes the ALNS algorithm used in the HALNS. First, the ALNS requires a starting solution s . s is either obtained by randomly applying insertion operators until every customer is served (first generation) or simply obtained from the population (further generations). The local best solution s^* is set to s (line 1 of Alg. 7). The ALNS makes use of the global best solution \hat{s} . Note that there is no global best solution in the first generation. This means that all solutions are equivalent at the beginning of the first generation, i.e., $\hat{s} = s^* = s$.

The ALNS generates a new solution s^{new} by iteratively removing and then inserting customers. The search proceeds while the number of iterations without finding a new best solution is lower than it^{stop} (line 2 of Alg. 7). At the beginning of every iteration, removal and insertion operators

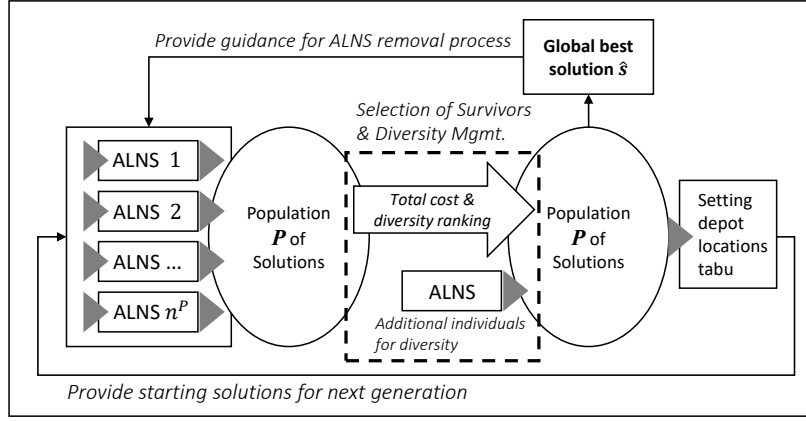


Figure 3.4: Schematic overview of a single HALNS iteration

Algorithm 7: ALNS algorithm in simulated annealing framework

Input : Starting solution s , global best solution \hat{s}
Output: best solution s^*

```

1  $s^* \leftarrow s$ 
2 while Iterations without improvement <  $it^{\text{stop}}$  do
3   ChooseOperators()
4    $C_R^C \leftarrow \text{getRemovalCandidates}(s, \hat{s})$  // Removal candidates (3.4.1.2)
5    $(s^{\text{new}}, C_R) \leftarrow \text{Remove}(s, C_R^C)$  // Removal operators (3.4.1.3)
6    $C_R \leftarrow \text{sort}(C_R)$  // Insertion order (3.4.1.4)
7    $s^{\text{new}} \leftarrow \text{Insert}(s^{\text{new}}, C_R)$  // Insertion operators (3.4.1.5)
8    $s^{\text{new}} \leftarrow \text{FirstLevelHeuristic}(s^{\text{new}})$  // Determine first-level solution (3.4.1.6)
9   if  $f(s^{\text{new}}) < f(s^*)$  then
10     $s^{\text{new}} \leftarrow \text{LocalSearch}(s^{\text{new}})$  // Local search (3.4.1.7)
11     $(s, s^*) \leftarrow \text{Symmetries}(s^{\text{new}})$  // Account for symmetries (3.4.1.8)
12    if  $f(s^*) < f(\hat{s})$  then
13       $\hat{s} \leftarrow s^*$ 
14  else if accept( $f(s^{\text{new}}), f(s^*), \tau$ ) then
15     $s \leftarrow s^{\text{new}}$  // Simulated annealing (3.4.1.9)
16   $\tau \leftarrow \tau \cdot \alpha$ 
17  UpdateParameters() // Adaptive parameters (3.4.1.10)
    
```

are chosen (line 3 of Alg. 7). The probability of choosing removal and insertion operators depends on the historic performance of operators. Next, a set of customers who are candidates for removal, C_R^C is generated by comparing the current solution s and the global best solution \hat{s} (line 4 of Alg. 7). The basic idea results from the observation that a certain number of edges in a good solution is similar to an even better solution. Customers are therefore to be added to C_R^C if they seem oddly placed in s compared to \hat{s} . The removal operator chosen then removes some or all customers from the solution who are included in set C_R^C (line 5 of Alg. 7). Data collected during the search determines the order in which the removed customers C_R are inserted (line 6 of Alg. 7). Next, the chosen insertion operator inserts all customers from the sorted set of previously removed customers C_R into the solution (line 7 of Alg. 7). After inserting all customers, the first-level costs have to be determined via a first-level heuristic (line 8 of Alg. 7).

Whenever a new best solution is found, a local search procedure is used to improve the routes (line 10 of Alg. 7). Before a solution is accepted, it must be checked for symmetries (lines 11 and

15 of Alg. 7)). The ALNS uses simulated annealing to escape local optima (lines 9-15 of Alg. 7). Please note that determination of the initial temperature is instance-specific. At the end of every iteration the temperature τ is multiplied by the cool rate α (line 16 of Alg. 7) and the parameters are updated (line 17 of Alg. 7).

Before we detail the parts of the ALNS in Subsections 3.4.1.2 - 3.4.1.10, we introduce the information collection via insertion operators.

3.4.1.1 Information Collection

The insertion operators collect data when inserting customers which is then used during the subsequent removal and insertion operations. These are

- the historic penalized costs C_j^{hist} ,
- the minimal encountered costs C_j^{min} ,
- and the number of times customer j has been reinserted, n_j .

C_j^{hist} is increased on every insertion of customer j by $\Delta_j = f(s_1) - f(s_0)$, where s_1 represents the solution after inserting customer j and s_0 the solution before inserting customer j . Thus, C_j^{hist} measures the accumulated increase in costs when inserting customer j . Similarly, C_j^{min} quantifies the minimum insertion costs of customer j found so far. n_j is simply increased by one if customer j has been reinserted.

3.4.1.2 Determination of Removal Candidates

Before applying one of the removal operators the set of removal candidates is determined by comparing the current solution s with the global best solution \hat{s} . We implement two variants for selecting which customers are added to set C_R^C corresponding to the location and the routing decisions considered. The first variant uses the routing decision, more specifically a customer is added to C_R^C if its successor in s differs from its successor in \hat{s} . Customers who do not fulfill this criterion may still be added to the set with probability p^{binom} . The second variant compares the location decision of customers. A customer who is served from a different location in s compared to \hat{s} will be added to C_R^C . The same probability-based approach as for the first variant is used. Furthermore, we use a third alternative to foster diversification, where $C_R^C = C$, i.e., there exists no pre-selection of customers. One of the three alternatives is randomly chosen in every iteration. The probability depends on the performance and is adapted during the search (see Section 3.4.1.10).

3.4.1.3 Removal Operators

The maximum number of customers to be removed in one iteration q^{binom} is sampled from a binomial distribution with sample size $|C|$ and probability p^{binom} . The number of customers that are actually removed by one of the removal operators below is expressed by $q^r = \min\{q^{\text{binom}}, |C_R^C|\}$. Thus, removal operators select q^r customers from set C_R^C according to the criterion specific to the removal operator used (e.g., randomly or according to costs), and then remove these customers from the current solution. The customers removed are added to set C_R .

Random Removal The operator selects and removes customers randomly.

Historic Cost Removal The operator selects customers in decreasing order of average historic penalized costs, i.e., in decreasing order of $\frac{C_j^{\text{hist}}}{n_j}$.

Worst Cost Removal The operator calculates the change in costs Δ_j if customer j is removed from the current solution. Customers are selected in descending order of $\Delta_j - C_j^{\text{min}}$.

Shaw Removal The operator selects customers similar to each other. The first customer is removed with the *Random Removal Operator*. The following requests are chosen from C_R^C in increasing order of relatedness. The relatedness $Rel(c_1, c_2)$ of two customers c_1 and c_2 is measured by the distance and demand of both customers. Customers with a lower $Rel(c_1, c_2)$ are more related and removed first.

$$Rel(c_1, c_2) = \frac{c_{c_1, c_2}}{\max_{i, j \in N} c_{ij}} + \frac{|d_{c_1} - d_{c_2}|}{\max_{j \in C} d_j} \quad (3.1)$$

Depot Removal The operator randomly selects a depot and removes either all customers who are served via this depot or randomly up to q^{binom} customers. Note that for this and the following operator, we set $C_R^C = C$, i.e., all customers are available for removal. This operator helps to close a depot if only few customers are served via this depot.

Least Efficient Vehicle Removal This operator uses the whole customer set for removal similar to the *Depot Removal Operator*. The operator identifies the least efficient vehicles. The least efficient vehicle has the highest ratio of tour costs compared to the number of customers on the tour, i.e., the highest cost per customer. The operator then removes customers who are served via this vehicle. The operator repeats the process until q^{binom} customers have been removed. If q^{binom} exceeds the number of customers served via the least efficient vehicle, for example, customers from the second least efficient vehicle are removed, and so on, until the total number of customers removed exceeds q^{binom} .

3.4.1.4 Determination of Insertion Order

The set of removed customers, C_R is sorted in descending order of average historic costs $\frac{C_j^{\text{hist}}}{n_j}$. This means that customers with high historic costs are inserted earlier, and there are therefore more options to insert them at a low-cost position. Customers with low average historic costs can be inserted later as they supposedly have more options to be inserted at an appropriate position with low costs.

3.4.1.5 Insertion Operators

After sorting C_R , the customers are inserted using one of the operators described in the following. The operators have to decide which depot the customer is to be served from and at which position in a route the customer is placed. Obviously both decisions influence not only each other but

also the decisions taken for customers inserted later. For example, if a depot is opened because a customer is inserted, the customers to be inserted in the following will more likely be assigned to that depot, instead of a new depot being opened. We therefore use variants of the *Best Insertion Operator* that neglect specific cost components. The total costs consist of costs when opening a depot (called depot costs), vehicle costs when using a previously unused vehicle, and distance costs when traveling from one node to the next. All operators include distance costs. Please note that the depot costs can be determined accurately for the LRP. In contrast, for the 2E-VRP we can only approximate depot costs as it would be extremely time-consuming to solve the first level-routing every time a customer is inserted. The depot (satellite) costs are approximated for the 2E-VRP using the approach of Mühlbauer and Fontaine (2021).

Best Insertion Operator The operator iterates across all depots and vehicles to find the position where the customer can be inserted with lowest total cost, i.e., the sum of depot costs, vehicle costs and distance costs. This operator favors positions in depots and vehicles that have already been opened.

Best Insertion Operator - without Vehicle Costs This operator neglects vehicle costs, so new vehicles may be used when the distance costs are lower in a new vehicle compared to vehicles already in use. When respecting vehicle costs, the difference in distance cost must be at least as high as the cost of using a vehicle, otherwise we have to accept high distance costs in favor of not using a new vehicle. This operator still favors depots that have already been opened.

Best Insertion Operator - without Depot Costs This operator neglects depot costs. A depot may be opened if the distance costs plus vehicle costs at the unopened depot are lower compared to a position within a used/unused vehicle at a depot that is already open. This operator favors the opening of additional depots and therefore helps to investigate further depot configurations.

3.4.1.6 First-Level Heuristic

The first-level problem is solved from scratch after all customers have been inserted in the second level, i.e., once the assignment of customers to satellites is known. The first-level heuristic is inspired by Hemmelmayr et al. (2012) and Mühlbauer and Fontaine (2021). The heuristic works in three steps. 1) If the demand of a satellite exceeds the capacity of the first-level vehicle, the heuristic creates back-and-forth trips until the remaining demand fits into one truck. 2) The heuristic generates an initial solution by inserting all open satellites (with their respective remaining demands) at its best position in a giant tour and then splitting the giant tour. We follow the giant tour as long as the capacity of the vehicle suffices. As soon as the capacity limit is reached we randomly decide with probability p^{split} whether we either split the demand (generating a split satellite) and return to the main depot or return directly to the main depot without the demand of the satellite that caused the capacity to be exceeded. 3) An improvement phase searches for better solutions, stopping when no more improvements can be found. Let r_i and r_j be a pair of first-level routes. The improvement phase uses a relocate and a swap operator to find improvements for all

pairs of first-level routes. The relocate operator relocates a satellite in r_i to a different position in the same route. The swap operator swaps two satellites s_1 in r_i and s_2 in r_j . The swap operator checks whether the capacity of the first-level vehicles suffices. If the capacity suffices on both first-level vehicles, both satellites are swapped. If the capacity only suffices for s_1 on r_j , s_1 is moved to r_j and s_2 is moved to an empty route, and vice versa. Steps 2) and 3) are repeated $10 \cdot n_s^{\text{open}}$, whereas n_s^{open} is the number of satellites in use. Please note that the heuristic is not needed if $n_s^{\text{open}} \leq 2$, as the optimal solution can easily be obtained. We randomly set $p^{\text{split}} = \text{unif}(0, 1)$ for each iteration. To speedup the procedure, a hash table keeps track of satellite configurations already examined. The first-level problem consists of only few satellites, so that this simple heuristic finds good solutions within reasonable runtimes.

3.4.1.7 Local Improvement of Routes

The local search procedure is used every time a new best solution is found to further enhance the quality of that solution. The local search procedure consists of simple insertion and swap operators working on the established routes of a solution. The local search maintains the location decision and route assignment of customers, resulting in pure intra-route improvement. We use the following operators, whereas c_1 denotes a customer, c_2 its successor and c a customer on the same route. The local search iterates in that order of operators across each customer c_1 and c in that route until no more improvement can be found. The first improving move is accepted.

- Insertion 1: Remove c_1 , then insert it after c .
- Insertion 2a: Remove c_1 and c_2 , then insert c_1 and c_2 after c .
- Insertion 2b: Remove c_1 and c_2 , then insert c_2 and c_1 after c .
- Swap 1: Swap c_1 with c .
- Swap 2: Swap c_1 and c_2 with c .

3.4.1.8 Accounting for Symmetries

After inserting every customer and executing the local search in case of a new best solution, the routes have to be aligned to account for symmetrical solutions. This is important as the determination of removal candidates (Section 3.4.1.2) and the diversity measure (Section 3.4.3) both rely on the successor relation of nodes. The direction of the route does not alter the cost, but a different direction is represented by a completely different successor vector. Same solutions would be identified as completely different. Consider an example with depot 0, customers 1 and 2 and two solutions A and B, solution A with route (0-1-2-0) and solution B with route (0-2-1-0). We can simply change the direction of the solution B and see that it is the same as solution A. The successor vector of the two customers 1 and 2 of solution A, however, is (2,0), and for solution B (0,1). A simple way to remove this kind of symmetrical solutions is to allow only routes where the first customer has a lower index than the last customer on that route (or the same index if there is only one customer on the route). This means the route of solution A is aligned correctly, whereas the route of solution B has to be inverted.

3.4.1.9 Simulated Annealing

Simulated annealing is used as an acceptance criterion. Deteriorating solutions are accepted with a probability depending on the difference in costs of the candidate solution $f(s^{\text{new}})$, the cost of the best solution obtained during the run of the ALNS $f(s^*)$ and the current temperature τ . A worse solution is accepted if $e^{\frac{-(f(s^{\text{new}})-f(s^*))}{\tau}} > \text{unif}(0,1)$.

The temperature τ is determined for every instance with $\tau = -\frac{\overline{\Delta E}}{\ln(\chi_0)}$ using the formula from Johnson et al. (1989) at the beginning of each ALNS run. $\overline{\Delta E}$ estimates the cost increase of strictly positive transitions and χ_0 expresses the probability of accepting a deteriorating solution. We execute n_0 iterations of the ALNS in order to generate the transitions. The temperature is reduced by α after each iteration.

3.4.1.10 Adaptive Parameters

The HALNS adaptively changes three kinds of parameters, which are (1) the probability of choosing operators (determining the removal candidates, removal and insertion operators), (2) the probability of the binomial distribution p^{binom} determining the number of requests to be removed, and (3) the weight ω for penalties.

Operators The probability of an operator depends on the historic performance, expressed by a score that is increased by either σ_1 , σ_2 or σ_3 (Ropke and Pisinger 2006). If a new best solution is found, the score is increased by σ_1 . If a previously unknown solution with lower costs than the current solution is found, the score is increased by σ_2 . If the solution has higher costs but is accepted through the simulated annealing procedure, the score is increased by σ_3 .

Probability of Binomial Distribution The probability of binomial distribution is adapted in a similar manner with $p^{\text{binom}} = \gamma \cdot \overline{p^{\text{binom}}} + (1 - \gamma) \cdot p^{\text{binom}}$. Parameter γ denotes the reaction factor and $\overline{p^{\text{binom}}}$ the average of the share of actual removed customers weighted by σ_1 , σ_2 or σ_3 . If in iteration A, for example, 20% of customers were removed and a new best solution is generated, and in iteration B 40% of customer were removed and the solution was accepted via simulated annealing, then $\overline{p^{\text{binom}}} = \frac{\sigma_1 \cdot 0.2 + \sigma_3 \cdot 0.4}{\sigma_1 + \sigma_3}$.

Weight for Penalties The weight for penalties is adapted after each generation depending on the number of infeasible solutions in the population. If the number of infeasible solutions within the generation is smaller than $\frac{n_p}{3}$ the weight is divided by five or if the number of infeasible solutions exceeds $\frac{2n_p}{3}$ the weight is multiplied by five.

3.4.2 Crossover and Education Phase

The crossover and education phase tries to combine two individuals by using the ALNS described previously instead of using an explicit crossover operator, which usually tries to combine individuals by maintaining good parts of these solutions. The ALNS uses solutions of the population one by one as a starting solution and the global best solution to guide the removal process. Customers are

removed if they differ in both solutions with respect to location assignment or successor assignment, as described in Section 3.4.1.2. This means the probability of removing customers who are already well placed decreases.

3.4.3 Selection of Survivors and Diversity Management

The selection of individuals is based on total costs and a diversity measure. We assign a cost rank $Rank_i^{\text{Costs}}$ to each individual i . Similarly, a diversity rank is assigned to each individual $Rank_i^{\text{Diversity}}$. The cost rank is determined from sorting the population in increasing order of total costs. The diversity rank is determined by sorting the population based on the hamming distance to all other individuals. The hamming distance is calculated by comparing the successor of nodes, $Succ^{\text{ind}}$ against the $Succ^i$ of all other individuals $i \in P$.

$$Hamming^{\text{ind}} = \sum_{i \in P} \sum_{j \in C} Succ_j^{\text{ind}} \neq Succ_j^i \quad (3.2)$$

Finally, the population is ordered in increasing order of the overall rank ($Rank_i^{\text{Overall}} = Rank_i^{\text{Costs}} + Rank_i^{\text{Diversity}}$).

Individuals with the n^P lowest overall ranks are used in the next generation. The population may grow until $|P| = 4n^P$. After reaching this point, excess individuals with the highest overall rank are removed from the population. New individuals are generated to foster diversification after each fifth generation.

3.4.4 Setting Depot Locations Tabu

After every generation, depot locations may be set tabu for the following generation. The procedure sets all depot locations tabu and then iterates across all individuals to check which depot locations are in use for that specific individual. As soon as a depot location is used for at least one individual, its use is permitted in the next generation. To further explore depot location configurations, depot locations may be set as non-tabu with a 30% probability. Similarly, depot locations may be set tabu with a 10% probability as long as the remaining total capacity suffices to accommodate the demand. This search strategy significantly reduces runtime and intensifies the search around promising depot location configurations.

3.5 Numerical Experiments

In this section we introduce the instances used and our experimental setting (Section 3.5.1), state the parameter tuning (Section 3.5.2), evaluate the performance of the HALNS on benchmark instances against approaches from literature (Section 3.5.3), and lastly evaluate the various components of the HALNS (Section 3.5.4).

3.5.1 Instances and Experimental Setting

Table 3.2 shows the instances used for the problem classes 2E-VRP, LRP and MDVRP and the respective sources for obtaining the instances. Please note that we use the MDVRP instances with a tight fleet size limit as given in <https://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/>. As pointed out by Sadati et al. (2021) there also exists a variant with a higher fleet size limit corresponding to the original values given by Cordeau et al. (1997).

Table 3.2: Instances

| Problem | Instances | Exemplary Source |
|----------------------|---|---|
| 2E-VRP ($n = 207$) | Set 2a, 2b, 2c Set 3a, 3b, 3c Set 4a, 4b Set 5 Set 6a | Breunig et al. (2016) http://www.univie.ac.at/prolog/research/TwoEVRP |
| LRP ($n = 79$) | Prodhon Tuzun Barreto | Prins et al. (2004) http://prodhonc.free.fr/Instances/instances_us.htm Tuzun and Burke (1999) http://prodhonc.free.fr/Instances/instances_us.htm Barreto et al. (2007) http://sweet.ua.pt/sbarreto/ |
| MDVRP ($n = 33$) | Cordeau | Cordeau et al. (1997) https://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-instances/ |

The HALNS is coded in C++. Experiments in Section 3.5.2 and 3.5.4 are conducted on an AMD Ryzen 7 2700X CPU with eight cores and 16 GB of RAM and in Section 3.5.3 on an AMD Ryzen 9 3900X CPU with twelve cores and 32 GB of RAM. All experiments use one thread. We use the same parameter setting for all experiments and problem classes after extensive preliminary parameter testing (see Section 3.5.2). We only vary the maximum number of generations, gen^{\max} , which is increased from $gen^{\max} = 10$ in Section 3.5.2 to $gen^{\max} = 100$ in Section 3.5.3 and $gen^{\max} = 50$ in Section 3.5.4. All parameters can be found in B.1.

To make the runtimes comparable we standardize runtimes by the passmark single thread rating of the CPUs used. The AMD Ryzen 9 3900X has a passmark single thread rating of 2731, while the AMD Ryzen 7 2700X has a rating of 2439. We can thus expect a runtime of the HALNS of 89.3% when running on 3900X, instead of running on 2700X. This approximation is reasonably accurate, as we show in B.2. In the following, all runtimes are standardized as if run on the AMD Ryzen 9

3900X. The respective passmark single thread ratings for all CPUs used in the relevant benchmark literature can be found in Table B.3.

3.5.2 Parameter Tuning

We tune a set of parameters that showed a relevant impact in the numerical study of Voigt et al. (2021) (see Table 3.3). These parameters are quantified as follows. First, we randomly select 30 instances from the total set of 2E-VRP instances. Starting from the initial parameter setting as given in Voigt et al. (2021), we alter the parameters one by one in the sequence and the range listed in Table 3.3. The initial values are marked with a superscript i .

Table 3.3: Parameter tuning for HALNS on 2E-VRP instances

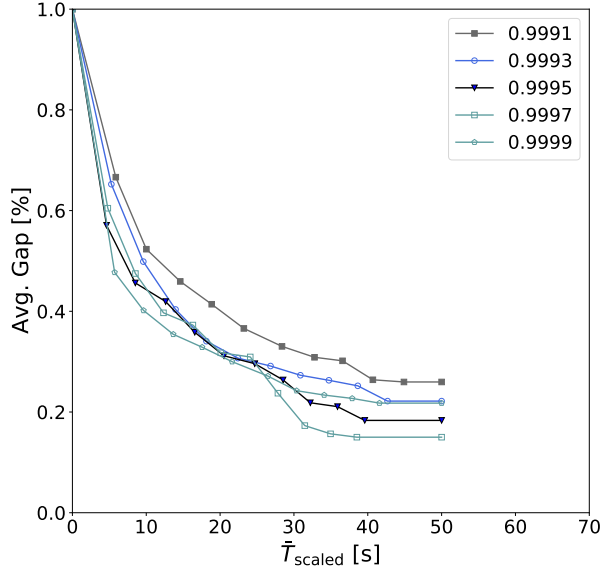
| Parameter | | Values | Chosen |
|--------------------|--|--|---------------------------|
| n^P | size of the initial population | 4, 12, 24, 36, 48 ^{i} | 12 |
| it^{stop} | number of ALNS iterations without improvement | 1000, 10000, 20000, 30000 ^{i} , 40000 | 10,000 |
| p^{binom} | probability for binomial distribution drawn at the beginning of every ALNS run | $\text{unif}(0.12, 0.24)^i$, $\text{unif}(0.10, 0.35)$, $\text{unif}(0.24, 0.48)$ | $\text{unif}(0.10, 0.35)$ |
| α | cool rate in SA | 0.9991, 0.9993, 0.9995, 0.9997, 0.9999 ^{i} | 0.9997 |
| χ_0 | acceptance probability in SA | 0.01, 0.10, 0.15, 0.25 ^{i} , 0.35 | 0.10 |

We run the HALNS five times for every parameter setting on the subset selected and set the number of generations to $gen^{\text{max}} = 10$ to reduce the computational effort. We plot the average search trajectory, i.e., the average gap achieved after each generation (with its corresponding average runtime) to assess the trade-off between runtime and solution quality. Lastly, upon visually inspection, we chose the value with the lowest curve, where the HALNS produces the same results within shorter runtime or better results within the same runtime. In other words, a parameter setting dominates, if the area under the respective search trajectory is the smallest.

As an example, Figure 3.5 illustrates the procedure for parameter α . The figure shows the search trajectories for the five tested values. Setting $\alpha = 0.9997$ yields the best result for longer runtimes. For shorter runtimes there is no clear best parameter. Similar results are found for the other parameters. The HALNS is therefore robust and not very dependent on parameters as long as the chosen values are within a reasonable range.

3.5.3 Benchmarks

We evaluate the performance of approaches on benchmark instances and average the results over all benchmark instances of one problem class. In the following tables, the first column shows the *Authors*, while the second column shows the *Approach* used. The *Best Gap* column represents the gap of the best run compared to the (previous) BKS, whereas column *Avg. Gap* represents the average gap over n runs. Most authors use five runs for 2E-VRP and LRP ($n = 5$), and ten ($n = 10$) for MDVRP. As a result, the HALNS is run five and ten times. The column $\bar{T}_{\text{scaled}} [s]$ represents the average standardized runtimes across n runs in seconds, as if run on the AMD Ryzen


 Figure 3.5: Search trajectories for different α on subset of 2E-VRP instances

9 3900X using the passmark single thread ratings, as described previously. The last column n represents the number of runs. The approaches are sorted according to their *Best Gap* performance. Furthermore, Figure 3.6-3.8 illustrate the average gaps for standardized runtimes to facilitate an easier comparison of approaches. The graphs of the HALNS are generated by varying the stopping condition, $gen^{\text{stop}} = [1, \dots, 50]$. In contrast to these figures, Table 3.4-3.6 show only the results when $gen^{\text{stop}} = 2, 10, \text{ or } 50$. For example, *HALNS 2* means that the HALNS stops after two generations without improvement, i.e., $gen^{\text{stop}} = 2$. Please note that the figures show only approaches with an average gap lower than 0.5%.

3.5.3.1 2E-VRP Benchmark

Table 3.4 shows the results for the 2E-VRP. Figure 3.6 additionally illustrates the trade-off between solution quality and computation time. The HALNS forms the efficient boundary, i.e., the HALNS dominates all other approaches both in terms of solution quality and computation time. *The HALNS 2* already generates good results with a best gap of 0.05% and an average gap of 0.16% within a short runtime of 27s. Only the approaches of Wang et al. (2017) and Amarouche et al. (2018) achieve a similar best gap, and while their average gap is lower, \bar{T}_{scaled} is considerably higher. The HALNS's best gap can be further reduced to 0.03% by increasing gen^{stop} to 10. Additional generations do not further improve the best gap. However, the average gap decreases to 0.04% for the *HALNS 50* and approaches the best gap, showing that additional generations stabilize results. The same pattern can also be observed for the LRP and MDVRP. The HALNS outperforms all pure ALNS implementations, showing the value of hybridizing the ALNS. Note that several approaches omit some of the benchmark sets. We therefore additionally give the HALNS's average gap for the considered subsets. The detailed results for the *HALNS 50* can be found in B.3, Table B.4, where it is shown that the *HALNS 50* finds 190 of 207 BKS.

Table 3.4: Heuristics for 2E-VRP

| Authors | Approach | Best Gap | Avg. Gap | \bar{T}_{scaled} [s] | n |
|---------------------------------------|-----------------|--------------|--------------|-------------------------------|----------|
| Voigt et al. | HALNS 50 | 0.03% | 0.04% | 286 | 5 |
| Voigt et al. | HALNS 10 | 0.03% | 0.07% | 83 | 5 |
| Wang et al. (2017) | VNS + Math | 0.04% | 0.08% | 126 | 5 |
| Amarouche et al. (2018) | LNS + Math | 0.05% | 0.10% | 65 | 5 |
| Voigt et al. | HALNS 2 | 0.05% | 0.16% | 27 | 5 |
| Mühlbauer and Fontaine (2021) | Hybrid LNS | 0.07% | 0.19% | 49 | 5 |
| Breunig et al. (2016) | LNS | 0.09% | 0.17% | 118 | 5 |
| Yu et al. (2021) ¹ | ALNS | 0.16% | 0.32% | 156 | 5 |
| Hemmelmayr et al. (2012) ² | ALNS | 0.24% | 0.48% | 34 | 5 |
| Jie et al. (2019) ³ | ALNS + Math | 0.60% | 0.82% | 552 | 5 |
| Enthoven et al. (2020) ⁴ | ALNS | 0.61% | 0.97% | 63 | 5 |

¹ Set 6 not included. Avg. gap of *HALNS 50* on the subset: 0.04%.

² Sets 2c, 3c, 4a and 6 not included. Avg. gap of *HALNS 50* on the subset: 0.06%.

³ no CPU indicated. We assume a passmark score of 1500, similar to publications from the same year.

⁴ Sets 4a, 4b and 6 not included. Avg. gap of *HALNS 50* on the subset: 0.08%.

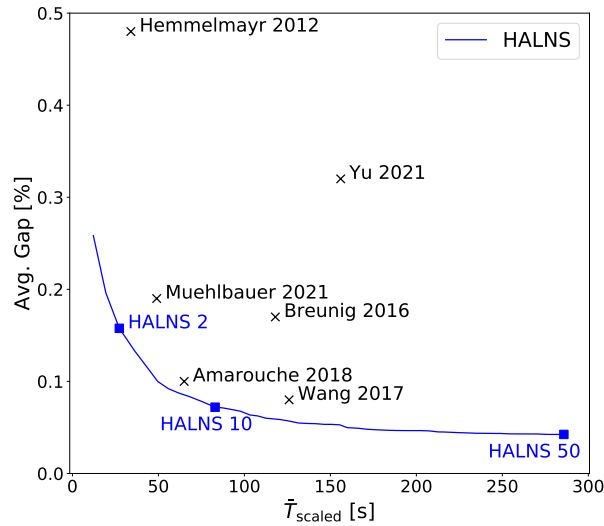


Figure 3.6: Performance chart 2E-VRP

3.5.3.2 LRP Benchmark

Table 3.5 and Figure 3.7 show the results for the LRP. Again, the *HALNS 2* already yields a good best gap of 0.16% at a low \bar{T}_{scaled} of 40s. Both best and average gap can be significantly improved by additional generations, up to 0.02% and 0.07%, respectively, which are the best results of all approaches considered. Again, additional generations reduce the difference between best and average gap and make the approach more robust. In total, we find 62 of 79 BKS (see B.3, Table B.5), including three newly found BKS, which are shown in detail in B.3, Tables B.7 - B.9. However, as easily seen in Figure 3.7, the approach of Schneider and Löffler (2019) with its different configurations achieves similar results as the HALNS. It dominates the HALNS for shorter runtimes. It is reasonable to assume that the approach of Schneider and Löffler (2019) shows a similar behavior as HALNS in the brackets between the given values, but for longer runtimes, the HALNS yields slightly better results.

Table 3.5: Heuristics for LRP

| Authors | Approach | Best Gap | Avg. Gap | \bar{T}_{scaled} [s] | n |
|------------------------------|------------------------|--------------|--------------|-------------------------------|----------|
| Voigt et al. | HALNS 50 | 0.02% | 0.07% | 389 | 5 |
| Schneider and Löffler (2019) | Tree + GTS (Quality) | 0.02% | 0.08% | 387 | 5 |
| Schneider and Löffler (2019) | Tree + GTS (Basic) | 0.04% | 0.16% | 109 | 5 |
| Voigt et al. | HALNS 10 | 0.05% | 0.15% | 121 | 5 |
| Voigt et al. | HALNS 2 | 0.16% | 0.37% | 40 | 5 |
| Schneider and Löffler (2019) | Tree + GTS (Speed) | 0.18% | 0.35% | 23 | 5 |
| Arnold and Sörensen (2021) | Progressive Filtering | 0.21% | 0.21% | 97 | 1 |
| Contardo et al. (2013) | GRASP + Math | 0.22% | 0.53% | 741 | 10 |
| Hemmelmayr et al. (2012) | ALNS | 0.45% | 0.79% | 94 | 5 |
| Lopes et al. (2016) | GA | 0.50% | 0.77% | 209 | 10 |
| Escobar et al. (2014a) | GTS + VNS | 0.66% | 0.66% | 40 | 1 |
| Ting and Chen (2013) | ACO | 0.79% | 0.79% | 22 | 10 |
| Escobar et al. (2013) | GTS | 0.93% | 0.93% | 77 | 1 |
| Yu et al. (2010) | SA | 0.96% | 0.96% | 237 | 1 |
| Duhamel et al. (2010) | GRASP + ELS | 1.09% | 1.09% | 182 | 5 |
| Prins et al. (2007) | GTS + Math | 1.47% | 1.47% | 3 | 1 |
| Prins et al. (2006a) | GA | 1.79% | 1.79% | 18 | 1 |
| Prins et al. (2006b) | GRASP + Path Relinking | 3.31% | 3.31% | 15 | 1 |

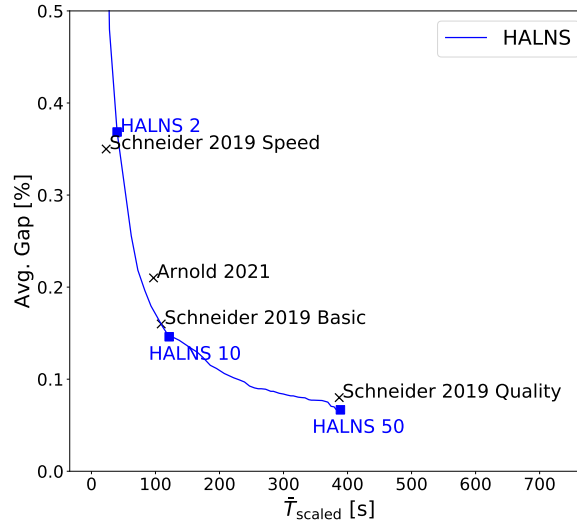


Figure 3.7: Performance chart LRP

3.5.3.3 MDVRP Benchmark

Table 3.6 and Figure 3.8 show the results for the MDVRP, detailed results are to be found in B.3, Table B.6. Overall, we find 27 of 33 BKS. Several approaches from literature deliver similar best gaps as the HALNS. The lowest best gap is achieved by the genetic algorithm of Vidal et al. (2012). The HALNS needs considerably more runtime to reach similar results compared to their approach and still produces a slightly worse best gap. However, the average gap is better. In contrast to that, Luo and Chen (2014) produce a slightly worse average gap than the HALNS, but a similar best gap at even lower runtimes than Vidal et al. (2012). Furthermore, the approach of Subramanian et al. (2013) delivers results similar to those of the HALNS, but within shorter runtimes. This slightly worse performance of the HALNS compared to the previous results on 2E-VRP and LRP

instances can be expected since the MDVRP differs most from the 2E-VRP. There is effectively only one insertion operator for the MDVRP. Furthermore, we expect that a stronger local search may be beneficial for solving the MDVRP, as used by Vidal et al. (2012).

Table 3.6: Heuristics for MDVRP

| Authors | Approach | Best Gap | Avg. Gap | \bar{T}_{scaled} [s] | n |
|---|---------------------|--------------|--------------|-------------------------------|-----------|
| Vidal et al. (2012) | GA | 0.00% | 0.09% | 52 | 10 |
| Voigt et al. | HALNS 50 | 0.01% | 0.05% | 695 | 10 |
| Luo and Chen (2014) | PSO | 0.01% | 0.3% | 17 | 10 |
| Subramanian et al. (2013) | ILS + Math | 0.02% | 0.08% | 364 | 10 |
| Voigt et al. | HALNS 10 | 0.03% | 0.12% | 187 | 10 |
| Voigt et al. | HALNS 2 | 0.06% | 0.22% | 63 | 10 |
| Arnold and Sørensen (2019) ¹ | Guided Local Search | 0.09% | 0.09% | 34 | 1 |
| Pisinger and Ropke (2007) | ALNS | 0.11% | 0.44% | 49 | 10 |
| Sadati et al. (2021) | TS + VNS | 0.2% | 0.3% | 297 | 25 |
| Cordeau and Maischberger (2012) | TS + ILS | 0.11% | 0.29% | 476 | 10 |
| Escobar et al. (2014b) | GTS | 0.19% | 0.19% | 41 | 1 |
| Renaud et al. (1996) | TS | 0.92% | 0.92% | 42 | 1 |
| Cordeau et al. (1997) | TS | 1.06% | 1.06% | 41 | 1 |
| Chao et al. (1993) ¹ | no classification | 2.74% | 2.74% | 37 | 1 |

¹ *pr* instances from Cordeau et al. (1997) not included. Avg. gap of HALNS 50 on the subset: 0.03%.

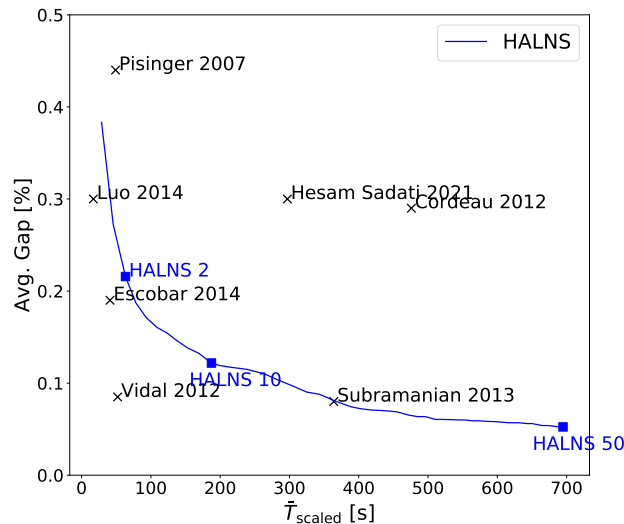


Figure 3.8: Performance chart MDVRP

3.5.4 Analysis of Algorithm Components

The HALNS combines a number of different heuristic components. The question therefore arises what influence each of these components has on the overall performance of the HALNS. We individually and independently disable the features and components listed below (see also Table 3.1) and compare the resulting performance to the performance of the full version of HALNS.

Hybridization: We disable the hybrid approach, i.e., neglecting the extended search of a population-based approach. In this case, the HALNS is reduced to a multi-start ALNS. The ALNS repeats $n_P \cdot \text{gen}^{\max}$ times without using information from previous generations (line 8 of Alg. 6). The

procedure disables the diversity management (line 10 of Alg. 6). The weights for penalties for infeasible solutions are not adapted. The depot locations are set tabu without considering information on depots that have already been used (line 11 of Alg. 6), which means the depot locations are just randomly set tabu.

Local improvement: We disable the additional local search procedure in case a new best solution is found during an ALNS run (line 10 of Alg. 7).

Simulated annealing: We exclude the diversification function within the ALNS. The ALNS then accepts only improved solutions (line 15 of Alg. 7).

Tabu search: We set depot locations within the search space never tabu (line 11 of Alg. 6).

Infeasible solutions: We increase the weight for penalties to a sufficiently high number, such that infeasible solutions are not accepted if the ALNS has already found a feasible solution.

Diversity management: We limit the diversity of the search space so that the surviving individuals are only selected according to their cost rank, $Rank_i^{Costs}$. Additionally, new individuals are not generated after each fifth generation (line 10 of Alg. 6).

Parameter adaptation: We disable the dynamic adaptation process of search space parameters. The probabilities for the choice of operators, the probability for the binomial distribution, p^{binom} , and the weight of penalties, ω , remain unchanged during the search (line 17 of Alg. 7).

Table 3.7 presents the results for all 2E-VRP instances when applying the HALNS algorithm with a certain configuration and when applying the HALNS disabling one mentioned component at a time.

Table 3.7: Analysis of HALNS solutions of 2E-VRP when disabling one component at a time

| Configuration | Best Gap | Avg. Gap | \bar{T}_{scaled} [s] |
|--|----------|----------|------------------------|
| HALNS | | | |
| $gen^{max} = 50, n^P = 12$ (full) | 0.03% | 0.04% | 204 |
| $gen^{max} = 1, n^P = 12$ (one generation) | 0.21% | 0.44% | 5 |
| $gen^{max} = 1, n^P = 1$ (one individual) | 0.47% | 1.27% | 1 |
| High-impact components | | | |
| Simulated annealing | 0.06% | 0.12% | 305 |
| Diversity management | 0.04% | 0.10% | 195 |
| Hybridization | 0.04% | 0.07% | 216 |
| Medium-impact components | | | |
| Tabu search | 0.03% | 0.06% | 224 |
| Parameter adaptation | 0.03% | 0.05% | 311 |
| Low-impact components | | | |
| Local improvement | 0.02% | 0.04% | 204 |
| Infeasible solutions | 0.02% | 0.04% | 205 |

Column *Best Gap* respective *Avg. Gap* shows the best and average gap from five runs. Column \bar{T}_{scaled} [s] shows the average standardized runtime as if run on the AMD Ryzen 9 3900X. The first

section of the table (HALNS) shows the results of the HALNS with $gen^{\max} = 50$, $n^P = 12$ (full), $gen^{\max} = 1$, $n^P = 12$ (one generation) and $gen^{\max} = 1$, $n^P = 1$ (one individual). The experiment with only one individual demonstrates that the ALNS on its own already generates reasonably good results with an average gap of 1.27% in under 1s of runtime. Executing just one generation with 12 individuals improves the average gap to 0.44%. The execution of the complete HALNS with 50 generations further stabilizes the results to an average gap of 0.04%, but increases the average runtime to 204s.

The following experiments apply the HALNS in the full configuration ($gen^{\max} = 50$, $n^P = 12$), and individually and independently disable one component. The second, third and fourth section of Table 3.7 (high-/medium-/low-impact components) show the respective results. The best and average gap deteriorates when high impact components are disabled compared to the full configuration. *Simulated annealing*, *diversity management* and *hybridization* highly impact the solution quality. Medium-impact components do not deteriorate the best gap when disabled, but deteriorate the average gap. *Tabu search* and *parameter adaptation* have a medium impact on the solution quality. Lastly, disabling *local improvement* and the absence of *infeasible solutions* do not affect the average gap and even slightly improve the best gap, while the effect on runtime is negligible. Note that multiple components not only improve solution quality, but also reduce runtime, e.g., *simulated annealing* and *parameter adaptation*.

Figures 3.9a, 3.9b and 3.9c show the solution gap as a function of scaled runtime for high-, medium-, and low-impact components, respectively. The graphs are generated by varying the stopping condition, just as in the previous section. The graphs are largely consistent across runtimes, i.e., a disabled component deteriorates the solution quality for high- and medium-impact components at all runtimes tested. The component *local improvement* (see Figure 3.9c), however, has a slightly different pattern. It seems to improve the performance for medium runtimes, but has no effect at long runtimes. The local search only improves routes (see Section 3.4.1.7). Its contribution to solution improvements is therefore rather limited at long runtimes.

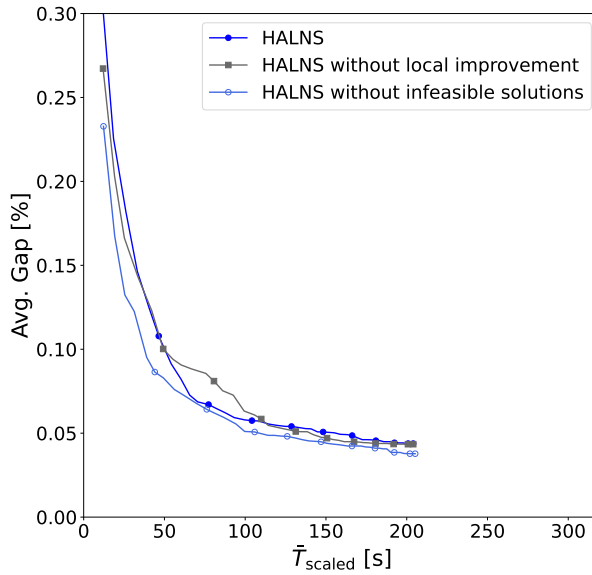
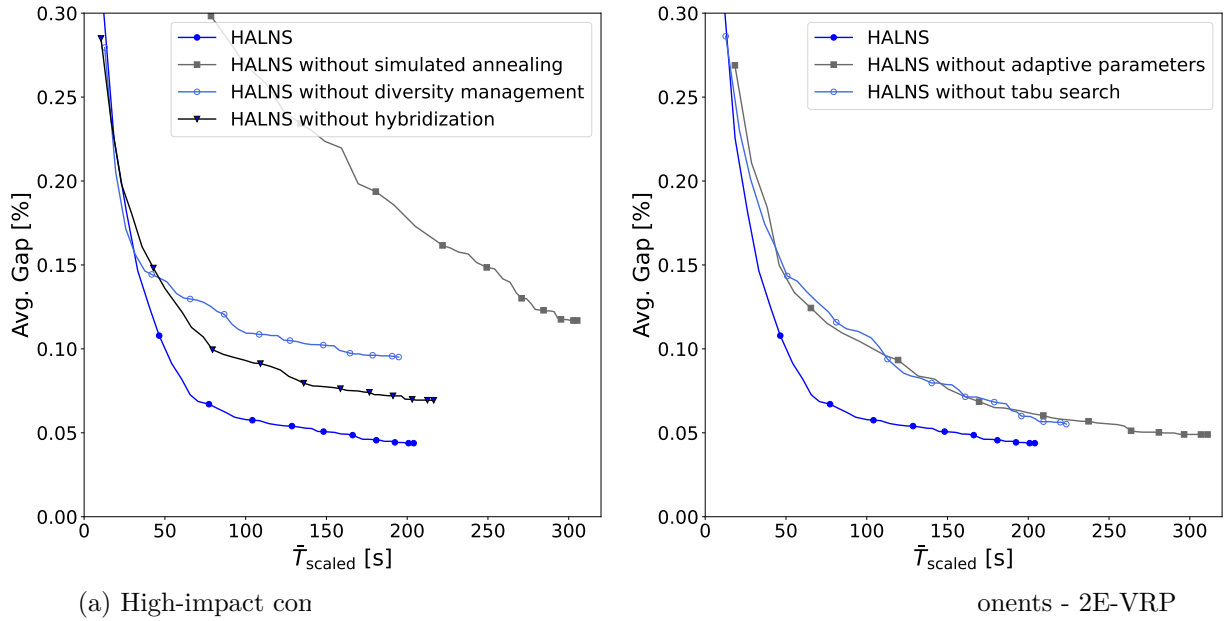


Figure 3.9: Analysis of components of HALNS for 2E-VRP

In light of the even slightly worse results when including *local improvement* and *infeasible solutions*, we reconstruct the experiment with MDVRP instances. Contrary to the results when solving the 2E-VRP, we find that the two components significantly impact the solution quality when solving the MDVRP (see Figure 3.10). The presence of infeasible solutions seems to be especially important when solving MDVRP instances, which include additional restrictions on the route length. This may be due to a well-fitting penalty weight ω for MDVRP instances or to characteristics of the instances, so that good solutions are close to infeasible solutions.

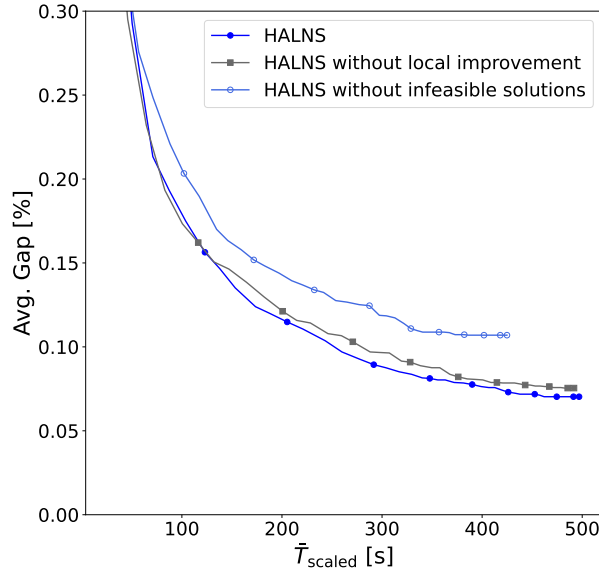


Figure 3.10: 2E-VRP low-impact components are high-impact components for MDVRP

In summary, the *simulated annealing*, *diversity management* and *hybridization* components strongly influence the solution quality when solving 2E-VRP instances. *Tabu search* and *parameter adaptation*, on the other hand, only have a medium influence on solution quality. However, our objective is a single solution approach for all the three problem classes and even components that have little effect on solution quality when solving 2E-VRP instances have a positive effect when solving other problem classes, i.e., LRP or MDVRP. All in all, we can conclude that a sophisticated combination of components is needed to improve solution quality at competitive runtimes for several problem classes of the 2E-VRP.

3.6 Summary and Further Areas of Research

Summary The design and operation of delivery systems also requires the determination of depots from which to deliver to customers. This general problem can be referred to as VRP with depot location decisions. We distinguish three variants of this problem class, the two-echelon VRP (2E-VRP), the location routing problem (LRP), and the multi-depot VRP (MDVRP). The literature offers a huge variety of quite specialized approaches for solving each of these problems.

We reviewed and classified these approaches and present a performance ranking. All three problems, however, can be solved with a single solution approach, since both the LRP and the MDVRP are special cases of the 2E-VRP. We developed a new metaheuristic method, the hybrid adaptive large neighborhood search (HALNS) based on the solution framework proposed by Voigt et al. (2021) that can solve all three problem classes. Computational experiments on benchmark sets from the literature demonstrate that the HALNS outperforms all approaches for the 2E-VRP and performs equally well with heuristics that are dedicated either to the LRP or the MDVRP. Furthermore, the HALNS shows robust results for all benchmarks, i.e., the differences between best and average runs are remarkably small. Additionally, the HALNS achieves three new best solutions

for LRP instances. Further, we have identified that different heuristic components of the HALNS with low performance impact when solving the 2E-VRP become relevant when solving LRP and MDVRP.

Further Areas of Research One main difficulty when inserting or removing customers during the search procedure is to approximate the impact of opening or closing a depot. Machine learning approaches could be used to approximate the resulting cost changes. Furthermore, the HALNS uses multiple ALNS runs to generate the population-based solution space. This structure offers the possibility of easy parallelization. The HALNS especially outperforms all pure ALNS implementations on these problem classes, demonstrating the value of an enlarged search space by using neighborhood- and population-based approaches. An interesting avenue for future research could therefore be the hybridization of existing and forthcoming ALNS approaches for other variants.

**Contribution 4 - A review on emerging variants
of the multi-period vehicle routing problem**

4 A review on emerging variants of the multi-period vehicle routing problem

Markus Frank

Abstract The extension of the classical Vehicle Routing Problem (VRP) to cover multiple periods is among the most common features within the VRP literature, due to the nature of many problem settings the VRP is applied to. Within the field of multi-period VRPs, especially applications with periodically recurring customer visits have been extensively studied. However, the strictly periodic VRP (PVRP) constitutes a special case of the more generic multi-period VRP (MPVRP). So far, the MPVRP in its non-periodic version has received less attention, although it arises in numerous applications, like dynamic problem settings and rich VRPs. Moreover, current literature misses a clear differentiation between MPVRP and PVRP as both terms are often used interchangeably. This paper provides a formulation for a general MPVRP and a comprehensive taxonomy for the recent literature devoted to periodic and non-periodic MPVRPs. As a field of special interest, it focuses on settings where the decisions on allowed visit periods and frequency are explicitly considered in the objective function.

Available at SSRN

URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4095035

4.1 Introduction

The well-known *vehicle routing problem* (VRP) seeks to group a given number of customers to tours and decide on the visiting sequence of each tour in a way that overall driven distance is minimized. It usually considers a restricted fleet of vehicles with a certain capacity needed to fulfill customers' demands and a single period only. Since its introduction, numerous publications investigated solution methods and various extensions.¹ This paper deals with one of the most studied extensions, which is its generalization to multiple periods. The feature of multiple periods needs to be addressed in many typical real-world applications of the VRP. For instance, online retailers often announce a latest delivery date for orders or their customers can choose a specific delivery day which needs to be respected for all customers in delivery tour planning (e.g., Archetti et al. 2015). Grocery stores require a supply of fresh goods a fixed number of times in a week, but deliveries should not be on consecutive days to ensure product availability throughout the whole week (e.g., Frank et al. 2021). In home healthcare services, patients may need help three times a week with the limitation that visits should have a minimum time lag of two days (e.g., Liu et al. 2014). Another example is given by Schrottenboer et al. (2018) where maintenance activities at offshore wind farms arise and need to be conducted before a deadline and technicians need to be assigned and routed accordingly.

It becomes apparent that problem settings can be differentiated by the characteristic of either periodically recurring visits or irregular visits, both possibly restricted by constraints on the allowable visit periods. The former variant is commonly known as the *periodic vehicle routing problem* (PVRP) which focuses on the strategic or tactical level, as for example delivery patterns of grocery stores are operated unchanged for several months, whereas the latter problem setting is mostly referred to as *multi-period vehicle routing problem* (MPVRP). It is typically used for tour planning in a more operational short-term planning horizon, but does not necessarily restrict to single visits only (e.g., Pasha et al. 2016). However, current literature lacks a clear differentiation and the terms PVRP and MPVRP are often used differently and interchangeably. The MPVRP clearly has its origins in the PVRP, but so far, no general problem formulation and model for the non-periodic MPVRP have been proposed.

In this paper we want to take a look at the various definitions of a “multi-period VRP”, propose a clear distinction to the PVRP and review publications that include a VRP setting with multiple periods, based on an appropriate typology and focusing on works in which the feature multi-period is of particular importance. The remainder of this paper is structured as follows. Section 4.2 describes the different understandings of the MPVRP in current literature, introduces the general problem setting and provides a corresponding general model. Section 4.3 presents a typology of MPVRP attributes to be used to structure the subsequent literature review. The recent literature of periodic and non-periodic VRPs with multiple periods are analyzed to obtain a clear view of the developments in the field. We conclude our findings and indicate future research directions in Section 4.4.

¹For a recent survey, see Vidal et al. (2020).

4.2 A Definition of the Multi-period Vehicle Routing Problem

When reviewing the current literature it becomes apparent that the terms MPVRP and PVRP are used inconsistently for VRPs with multiple periods. We therefore give a short overview of the different understandings to be found and propose a unified point of view to be able to differentiate between the two in Section 4.2.1. We then present a definitional model formulation for a general MPVRP in Section 4.2.2.

4.2.1 “PVRP” versus “MPVRP”

In order to find criteria when to use which term, the relevant differences between MPVRP and PVRP need to be identified. For this reason, we first recall the formal description of the PVRP as the older, more precisely defined variant and more consistently used term and then take a look at the occurring usage of the terms PVRP and MPVRP.

As stated in the survey by Campbell and Wilson (2014), the PVRP is defined as follows. The objective is to minimize total transportation costs by assigning each customer to one of its allowable visit patterns and create corresponding daily routes (clustering & routing decision). A visit pattern contains a subset of periods² whose combination is allowed for visiting the customer. Also, it is standard to define a visit frequency that gives the number of needed visits within the planning horizon. In most publications, the visit frequency is given additionally to the allowed visit patterns, but the visit frequency can also be the only constraint (e.g., Russell and Igo 1979). For a given visit frequency, the corresponding demands are typically fixed and equal for all visits or dependent on the spacing between two visits. For instance, if a customer is visited every three days, the demand for three days must be satisfied at each visit. Constraints include a given number of vehicles available on each day, with a limited capacity and a limitation on the total travel time, which start and end all of their tours at a single depot. Also, no customer demand should be greater than a vehicle’s capacity. Concerning the allowable visit patterns, Campbell and Wilson (2014) find three different variants for defining the set of feasible combinations.

- A predetermined set of allowable visit patterns is used.
- A fixed spacing between visits is given, e.g., a visit needs to be conducted every two days.
- A minimum and maximum spacing is given, which means the time between two visits is flexible to a certain extent.

Note that the latter two alternatives clearly state the periodicity of visits. Periodicity means that it is usually assumed that the last day of the given planning horizon is followed by the first one again and the created tour plan cyclically repeats. Even though the literature on the PVRP is not completely consistent regarding the problem definition, the term PVRP is used continuously since its introduction by Christofides and Beasley (1984) and Gaudioso and Paletta (1992). This is in contrast to the usage of the term MPVRP which does not refer to a specifically defined problem that has been introduced formally. In the next section, we provide a general MPVRP to close

²Note that we use the terms day and period interchangeably.

this gap. The term MPVRP is mostly used in the field of dynamic VRPs, where a number of publications considers the *dynamic multi-period vehicle routing problem* (DMPVRP) (e.g., Wen et al. 2010, Albareda-Sambola et al. 2014, Archetti et al. 2015) and in the field of rich VRPs, which are characterized by a great number of side constraints, very often including the multi-period feature (Lahyani et al. 2015). The mentioned features that differentiate a MPVRP from a PVRP are a visit frequency of one for each customer and that there is no periodicity in the service (e.g., Rothenbächer 2019, Archetti et al. 2015). This would define the MPVRP as a special case of the PVRP. However, both terms have been used differently.

- The term PVRP is used for all multi-period VRP settings, including those where visit frequency is equal to one and no periodicity is assumed.
- The term MPVRP is used for all multi-period VRP settings, including those with periodicity in visits (e.g., Estrada-Moreno et al. 2019).
- The term MPVRP is used for multi-period VRP settings in which each customer is visited exactly once and the term PVRP for settings with higher visit frequencies (e.g., Rothenbächer 2019).
- The term *tactical planning vehicle routing problem* (TPVRP) is used for multi-period VRP settings in which each customer is visited exactly once and the term PVRP for settings with higher visit frequencies (Baldacci et al. 2011).
- The term MPVRP is used as long as there is no periodicity in the service, regardless of visit frequency (e.g., Archetti et al. 2015).

Most works on the PVRP structure their input data quite strictly and assume certain periodical structures. The visit frequency is set to the same for all customers, all allowable visit patterns for each customer are non-overlapping, show equal intervals between two visits and every day is contained in at least one pattern (Rothenbächer 2019). Therefore the PVRP can not only be seen as a “VRP with [multiple] periods”, but has periodic demands as a core aspect. Visit frequencies of one are rare, but possible in the PVRP, especially in PVRP extensions which also decide on the visit frequency (e.g., Francis et al. 2006, Rothenbächer 2019, Frank et al. 2021). But weekly visits are also periodically repeated visits. Beyond that, there also exist non-periodic multi-period VRPs variants that consider multiple visits (i.e., a visit frequency higher than one) within the planning horizon (e.g., Pasha et al. 2016). That is why we propose to follow the point of view of Archetti et al. (2015) and define the MPVRP to be a special case of the PVRP where no periodicity is considered. Thereby, the MPVRP states the short-term planning problem (as opposed to the long-term PVRP), where the planning period is not meant to be continuously repeated, but finite. This means the MPVRP’s planning period ends at a certain point when used for short term planning only or it would be repeatedly solved in a rolling planning horizon, depending on the planning environment. As before, we use the term MPVRP for the non-periodic VRP with multiple periods, the term PVRP for the periodic VRP with multiple periods, and we speak of “VRPs with multiple periods” to address both and other problem classes with the feature multiple periods.

4.2.2 A General Multi-period Vehicle Routing Problem and its Formulation

As a foundation for discussing different problem variants in the following sections, we formulate a general MPVRP based on our findings above in which customers may be visited on every day and

transportation costs are to be minimized. We then show two of the most common extensions, i.e., visit period intervals and day-dependent service costs.

A general version of the MPVRP can be defined on a directed graph $G(N, A)$ with node set N and arc set A . Node set N consists of the depot 0 and the customer set C , $N = C \cup \{0\}$. The arc set is defined as $A = \{(i, j) : i \neq j, i, j \in N\}$. c_{ij}^{trans} denotes the associated transportation costs with each arc. Let $T = \{1, \dots, |T|\}$ be the planning horizon consisting of periods t . In each period, a given fleet of K delivery trucks with the same restricted vehicle capacity Q is available to perform routes starting and ending at the depot. $|K|$ states the maximum number of tours in each period. We define a customer to have a single demand d_i that can be fulfilled on any day (split-delivery is not allowed). If multiple visits at the same customer are needed, this can be modelled by co-locating customers with individual demands. Three decision have to be taken: (i) which days the customers are assigned to, (ii) the clustering of customers to tours and (iii) the sequence of customers on each tour. The following binary decision variables are applied:

- x_{ijkt} indicating whether arc (i, j) is traversed by vehicle k in period t ; $i, j \in N, k \in K, t \in T$.
- z_{ikt} indicates whether customer i is served by vehicle k in period t ; $i \in C, k \in K, t \in T$.

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

$$\text{Minimize TC} = \sum_{k=1}^K \sum_{i=0}^N \sum_{j=0}^N \sum_{t=1}^T c_{ij}^{\text{tran}} \cdot x_{ijkt} \quad (4.1)$$

s.t.

$$\sum_{k=1}^K \sum_{j=0}^N \sum_{t=1}^T x_{ijkt} = 1 \quad \forall i \in C \quad (4.2)$$

$$\sum_{i=1}^C d_i \sum_{j=0}^N x_{ijkt} \leq Q \quad \forall k \in K, t \in T \quad (4.3)$$

$$\sum_{j=0}^N x_{ijkt} = \sum_{j=0}^N x_{jikt} \quad \forall i \in N, k \in K, t \in T \quad (4.4)$$

$$z_{ikt} = \sum_{j=0}^N x_{ijkt} \quad \forall i \in N, k \in K, t \in T \quad (4.5)$$

$$\sum_{i,j \in S} x_{ijkt} \leq \sum_{i \in S} z_{ikt} - z_{jkt} \quad S \subset C, \forall j \in S, k \in K, t \in T \quad (4.6)$$

$$\sum_{j=0}^N x_{1jkt} \leq 1 \quad \forall k \in K, t \in T \quad (4.7)$$

$$\sum_{j=0}^N x_{j1kt} \leq 1 \quad \forall k \in K, t \in T \quad (4.8)$$

$$x_{ijkt} \in \{0, 1\} \quad \forall i, j \in N, k \in K, t \in T \quad (4.9)$$

$$z_{ikt} \in \{0, 1\} \quad \forall i \in N, k \in K, t \in T \quad (4.10)$$

The objective function (4.1) minimizes total costs, which in this case are overall transportation costs. Constraints (4.2) ensure that every customer is visited exactly once during the planning horizon. Constraints (4.3) prohibit the vehicle capacities to be exceeded on any day. Equations (4.4) conserve flow. Equations (4.5) link the assignment variables z_{ikt} with the flow variables x_{ijkt} . Constraints (4.6) are the subtour elimination constraints, formulated in terms of z_{ikt} variables. Equations (4.7) and (4.8) ensure that only one tour is performed per vehicle and day. Equations (4.9) and (4.10) define the variable domains. This model is a generalization of the VRP that is known to be an NP-hard optimization problem (see, e.g., Toth and Vigo 2014).

Extension by delivery intervals One of the most frequent extensions of the MPVRP is the introduction of delivery intervals that restrict the visit of each customer to specified, consecutive days (see, e.g., Bostel et al. 2008). This can be done by setting an earliest delivery day e_i and a latest delivery day l_i for each customer $i \in C$. Constraints (4.2) are then modified to allow exactly one visit within the delivery interval instead of the whole planning horizon.

$$\sum_{k=1}^K \sum_{j=0}^N \sum_{t=e_i}^{l_i} x_{ijkt} = 1 \quad \forall i \in C \quad (4.11)$$

Extension by service costs In its basic form formulated above, the MPVRP only takes transportation costs into account. It is therefore beneficial to visit customers close to each other on the same day and tour. However, service costs c_{it}^{serv} can be added (see, e.g., Baldacci et al. 2011) to reflect the common situation that visiting a customer i occurs costs depending on the chosen day t (or profits in the case of a negative c_i^{serv}). The objective function (4.1) is then complemented as follows.

$$\text{Minimize TC} = \sum_{k=1}^K \sum_{i=0}^N \sum_{j=0}^N \sum_{t=1}^T c_{ij}^{\text{tran}} \cdot x_{ijkt} + \sum_{k=1}^K \sum_{i=1}^C \sum_{t=1}^T c_{it}^{\text{serv}} \cdot z_{ikt} \quad (4.12)$$

The presented model can be easily adapted further to account for other common constraints and extensions, e.g., waiting or inventory costs, heterogeneous vehicles and multiple customer visits.

4.3 Literature Review

In this section, we provide a comprehensive literature review that focuses on VRPs with multiple periods. Based on our findings in Section 4.2.1, we distinguish between MPVRP and PVRP settings according to their assumptions on the periodicity of visits. We review MPVRPs in Section 4.3.1 first and PVRPs in Section 4.3.2 second and finally give an overview of used solution approaches in Section 4.3.3. In order to find reasonable structures for the reviews, we introduce a typology for the MPVRP problem class which is later adapted to PVRPs. Our paper retrieval procedure for the whole survey is based on an initial Web of Science search via keywords "vehicle routing problem with multiple periods", "multi-period vehicle routing problem" and "period[ic] vehicle routing problem", followed by a snowball search of literature cited by relevant publications.

4.3.1 Literature on the MPVRP

As mentioned throughout the paper, the term MPVRP was never formally introduced but merely separated from the PVRP over time to be used for non-periodic short-term planning. Many short-term problem settings feature some kind of dynamism which may be the reason for the wide-spread usage of the term MPVRP within the field of dynamic VRPs. Overall, there is a high number of publications considering MPVRPs, we therefore first present a typology of the MPVRP (Section 4.3.1.1) and then divide the characterization of its literature body into the two subsections of static (Section 4.3.1.2) and dynamic MPVRPs (Section 4.3.1.3).

4.3.1.1 A Typology for Multi-period Vehicle Routing Problems

The problem settings discussed in literature vary strongly, therefore it is no straightforward task to develop a comprehensive typology for the MPVRP. Some typologies including VRPs with multiple periods have already been suggested. For instance, Mourgaya and Vanderbeck (2006) present a classification for the PVRP focusing on different objective functions. Vidal et al. (2020) develops a typology for various VRP extensions. Mor and Speranza (2022) give an overview of “VRPs over time”, which also includes the basic PVRP. However, none of these publications addresses the attributes specific to the feature multiple periods and provides a comprehensive literature review on the MPVRP. After scanning the literature, we therefore present a typology based on multi-period-related attributes. In the following, we give a short explanation of each attribute.

Visiting frequency A first criterion for distinguishing MPVRP is the required number of visits at the customer. The multiple visits attribute is a necessary criterion for PVRPs (which we cover in Section 4.3.2). However, for MPVRPs, customers can be visited only once. Visits of customers in a PVRP with frequency one can be seen as multiple visits, because they are cyclically repeated, in contrast to a MPVRP with a finite planning horizon.

Allowable visit periods A second criterion arises with the feature of allowable visit periods, i.e., whether the set of periods in which a customer may be visited is restricted (see Section 4.2.1). If that is the case, the type of restriction can be further specified. The two possible types are hard constraints, where a visit on any other day than the allowed ones is not possible, and soft constraints which allow the violation of given visit restrictions at a cost. This includes setting the visit day to a day outside of the planning horizon, i.e., not delivering at all.

Objective function and visit period-related components Similar to Mourgaya and Vanderbeck (2006), we classify works according to their objectives considered. The three main streams can be divided in minimizing routing costs, maximizing the number of customers served and multi-objective formulations with application-specific objectives that do not coincide. As we focus our review to settings in which considering multiple periods plays a major role, it is reasonable to additionally indicate works in which the objective function includes a visit-period-related component, such as day-dependent service costs, penalty costs for late or no visit, waiting times or service

frequency, all directly determined by the decisions on when and how often to visit customers and independent from subsequent routing decisions. We distinguish between such models with an explicit visit period-related component and models where the decisions on frequency and specific visit days only have an implicit impact on the objective function, e.g., by influencing the routing costs or the number of vehicles to be used in a period.

All considered multi-period-related attributes for the MPVRP are summarized in Figure 4.1. The used attributes are explained in Table 4.1. In addition, we also indicate the most encountered VRP-related attributes in the literature characterizations below. They are defined in the respective sections.

Table 4.1: Legend for Tables 4.2-4.3

| Attribute | Abbreviation | Explanation |
|--------------------------------|--------------|--|
| Multiple Visits | MV | Checked, if customer requires multiple visits. If not checked, single visits are considered. |
| Visit periods | VP | H for hard constraints, S for soft constraints. |
| Objective function | Obj | R for routing costs (distance, time, transportation costs, etc.) FS for fleet size/fleet mix objectives NC for number of served customers O for other single-objective function MO for multi-objective function |
| Visit-period-related component | VC | Checked, if objective function contains an explicit visit-period related component such as day-dependent service costs |

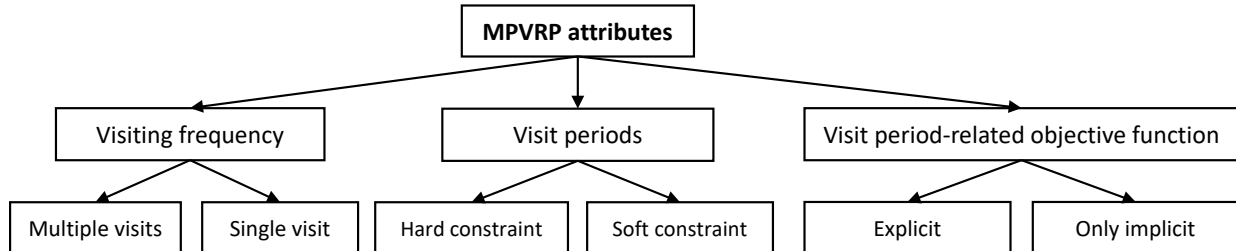


Figure 4.1: Attributes of MPVRPs.

4.3.1.2 Static MPVRP

First, we review the literature on static MPVRPs, which means all input data is known in advance, however, not necessarily with certainty. Overall, 14 publications can be attributed to this MPVRP type. A majority of papers (8) addresses specific applications. Table 4.2 shows the attributes of the reviewed works. In the following, we shortly describe the main contributions and indicate some interrelated streams of literature.

Baldacci et al. (2011) can be seen as first work on the MPVRP as the authors explicitly consider a variant of the PVRP for short-term planning which is referred to as TPVRP (see Section 4.2.1). However, their formulation already includes allowable visit intervals consisting of consecutive days

Table 4.2: Characterization of Static MPVRPs

| Paper | Application | Obj | MP-specific | | | VRP-related ^a | | | |
|---------------------------------|--------------|------|-------------|----|----|--------------------------|----|----|----|
| | | | VC | MV | VP | HF | TW | DC | WB |
| Baldacci et al. (2011) | Generic | R+O | x | | H | | | | |
| Pacheco et al. (2012) | Distribution | R | | | H | | | | |
| Athanasopoulos and Minis (2013) | Generic | R | | | H | x | x | x | |
| Dayarian et al. (2015) | Collection | R | | x | | | | | |
| Archetti et al. (2015) | Parcel | R+O | x | | S | | | | |
| Kim and Lee (2015) | Green VRP | R+O | | x | H | x | | | x |
| Schönberger (2016) | Collection | R | | | | | | x | x |
| Pasha et al. (2016) | Generic | FS+R | | x | H | | | | |
| Dayarian et al. (2016) | Collection | R | | x | | | | | |
| Cantu-Funes et al. (2018) | Distribution | R | | x | H | x | x | x | x |
| Estrada-Moreno et al. (2019) | Generic | R+O | x | | S | | | | |
| Larrain et al. (2019) | Generic | R+O | x | | S | | | | |
| Darvish et al. (2020) | Generic | R+O | x | | H | | | | |
| Muñoz-Villamizar et al. (2021) | Green VRP | MO | x | | S | | | | |
| Share in % | - | - | 50 | 36 | 79 | 21 | 14 | 21 | 21 |

^a HF: heterogeneous fleet, TW: time windows, DC: distance/duration constrained, WB: Workload Balancing

and service costs. By using a branch-and-bound algorithm, they are able to optimally solve instances up to 100 customers.

A static-stochastic VRP with multiple periods, but without delivery intervals is discussed by Dayarian et al. (2015). The problem arises in the collection of milk at local producers. The quantities to be collected are subject to weekly and seasonal fluctuations. Due to contractual agreements, a single vehicle routing plan must be designed for the whole planning horizon and not be changed afterwards. They solve the proposed model by a branch-and-price approach. Dayarian et al. (2016) develops an adaptive large neighborhood search in order to generate solutions for larger instances. Another stream is on the MPVRP with due dates (MVRPD). Motivated by a problem setting in city logistics, it is introduced by Archetti et al. (2015). Each customer demand is associated with an earliest and latest delivery date (i.e., release and due date). Orders may be postponed at a penalty cost. Additionally, inventory costs for each day an order stays at the depot after its release date incur. They also generate a first test instance set and solve it by using a branch-and-bound procedure. Building on their findings, Larrain et al. (2019) present two new solution approaches for the MVRPD, an optimized new branch-and-bound algorithm and a variable MIP neighborhood descent algorithm, which embeds a local search within a branch-and-bound procedure and clearly outperforms the previous approach. Muñoz-Villamizar et al. (2021) integrate ecological factors into the MVRPD. In order to investigate shipment consolidation strategies for green home delivery, three different objective functions are used.

Various other MPVRP variants are introduced. Pacheco et al. (2012) seek to design routes for delivering bakery products to satisfy the orders placed for the week, which are associated with deadlines. Athanasopoulos and Minis (2013) introduce the MPVRP with time windows and also consider a heterogeneous fleet, its size varying from period to period. They develop an exact solution method based on column generation. Schönberger (2016) consider a MPVRP with restricted capacity per period. Due to that restriction, quantities cannot be collected immediately but are

visited some periods later, however, within a specified deadline. Pasha et al. (2016) introduce the MPVRP including the fleet size and mix problem, where the demands of the customers vary between periods, and all customers may have to be visited in each period. Cantu-Funes et al. (2018) formulate an extensive MPVRP motivated by the case of a brewing company supplying DCs from production plants. In order to not run out of inventory, the DCs impose due dates to be served. Additionally, the model is characterized by a higher costs for using additional external vehicles, which can be seen as a variant of penalty costs. Kim and Lee (2015) extend the MPVRP by introducing a limit of carbon emissions per period. Trading costs for additional carbon emission certificates incur when this limit is exceeded, like a penalized soft constraint on the overall tour length per period. Estrada-Moreno et al. (2019) consider the MPVRP with price discounts for delivery flexibility. Customers indicate a single preferred delivery day which is not guaranteed by the service provider. Instead, the customer gets a fixed price discount if the delivery takes place on any other day than the preferred one.

4.3.1.3 Dynamic MPVRP

In the dynamic MPVRP, input data is not (fully) available when planning takes place, but gets (partially) revealed over time. As additional information arrives on every further day, re-planning actions with regard to the first tour plan are necessary, conducted either instantly as new information becomes available or periodically (e.g. daily), in a rolling horizon. Thereby, dynamic MPVRPs are usually applied on an operational planning level which may be the reason for widespread usage of the term MPVRP in this field (as opposed to the mid-term-oriented planning approach of the PVRP). Additionally, several works in this field also assume that some problem information is only available in a stochastic manner, as shown in Table 4.3. For complete reviews on the dynamic VRP, we refer to Psaraftis et al. (2016) and for dynamic and stochastic VRPs to Ritzinger et al. (2016) and Soeffker et al. (2021). In total, we can retrieve 17 publications on dynamic MPVRPs. However, we match some works together in the table due to identical model formulations.

Numerous publications focus the dynamic MPVRP including same-day delivery: Angelelli et al. (2007a) are the first to introduce the dynamic MPVRP (DMPVRP), where it is decided on each day which orders to fulfill on the current day and which orders to postpone without knowing the set of new requests that will be issued the day after. They consider a special case of delivery intervals where orders can be postponed only for one day, so every order has to be delivered either on the day the order arrived or the next one. Three simple algorithms are compared, one of them being optimal for the case of a two-day planning horizon and outperforming the other two for longer horizons. These algorithms are further analyzed in Angelelli et al. (2007b) for a special case. Furthermore, Angelelli et al. (2009b) builds on these model definition and findings and extend the problem setting by introducing a set of vehicles and a maximum route length constraint. Additionally, some orders may arrive during the day and tour plans can therefore be modified while vehicles are already traveling between customers. Angelelli et al. (2009a) proposes a rolling horizon solution framework and different short-term strategies, formulated as optimization models and solved by a variable large neighborhood search. Angelelli et al. (2010) tests additional numerical experiments based on the existing model. Ulmer et al. (2018) incorporates stochastic information,

Table 4.3: Characterization of Dynamic MPVRPs

| Paper | Application | Obj | MP-specific | | | VRP-related ^a | | | |
|--------------------------------|---------------|--------|-------------|----|----|--------------------------|----|----|----|
| | | | VC | MV | VP | ST | TW | DC | WB |
| Angelelli et al. (2007b,a) | Same-day | R | | | | | | | |
| Tricoire (2007) | Home services | NC+R | | | H | | x | x | |
| Bostel et al. (2008) | Home services | R | | | H | | x | x | |
| Angelelli et al. (2009a) | Same-day | NC+R | | | | | | | |
| Angelelli et al. (2009b, 2010) | Same-day | R+O | x | | S | | | x | |
| Wen et al. (2010) | Agriculture | R+O | x | | H | | | x | x |
| Albareda-Sambola et al. (2014) | Generic | R+O | x | | H | x | | | |
| Ninikas et al. (2014) | Parcel | R | | | H | | x | | x |
| Cordeau et al. (2015) | Auto-carrier | FS+R+O | x | x | H | | | | |
| Billing et al. (2018) | Auto-carrier | FS+R+O | | x | H | x | | | |
| Ulmer et al. (2018) | Same-day | NC | | | H | x | | x | |
| Bonassa et al. (2019) | Auto-carrier | R | x | x | H | | | | |
| Ulmer (2020) | Same-day | NC | | | H | x | | x | |
| Laganà et al. (2021) | Parcel | R | | | H | | | | |
| Subramanyam et al. (2021) | Gases | R | | | H | x | | | |
| Share in % | - | - | 29 | 18 | 94 | 29 | 18 | 41 | 12 |

^a ST: stochastic demand, TW: time windows, DC: distance/duration constrained, WB: workload balancing

as customer orders arrive during the day according to a known stochastic distribution. The problem is formulated by a markov decision process and solved by a time budgeting approach which anticipates the number and location of future orders intra- and inter-periodically by means of a value function approximation approach that simulates future realizations based on historic data. Finally, Ulmer (2020) define a new solution approach with the objective to combine the benefits of an online algorithm that is able to respond quickly in the dynamic same-day setting and those of an offline algorithm that has more computation time available to pre-determine possible outcomes. Both algorithms use value function approximation by simulation.

Concerning other applications, Wen et al. (2010) consider a case study involving the deliver of fodder to farmers, the authors formulate another extended variant of the DMPVRP. The objective function takes customer waiting costs and workload balancing into account. Albareda-Sambola et al. (2014) build on the deterministic model of Wen et al. (2010) and add the assumption that for each customer and day there is a probability given that the day is an allowed day, i.e. it is included in the customer's delivery interval. An adaptive policy is proposed to determine which customer to serve on each day and which to postpone.

Several papers are motivated by the delivery of produced vehicles to dealerships which are requested dynamically and with deadlines. In these problems, the set of customers is known and multiple visits to each one are assumed. Additionally, the loading of the vehicles on carriers is considered. Cordeau et al. (2015) propose an ILS, applied in a rolling horizon framework to capture the dynamic nature of the problem. For increased efficiency, Billing et al. (2018) use the historic orders of a customer to forecast future ones. Bonassa et al. (2019) focuses on finding the best combination of vehicles to be loaded into the carriers, permitting deadline violations at a cost.

Tricoire (2007) address a problem arising in the scheduling and routing of technicians. They encounter two kinds of demands, which must either be served at a specific day at any cost or can

be scheduled flexibly within a given number of periods. Bostel et al. (2008) study a similar application, however embedded in a rolling horizon planning framework. In a related manner, Ninikas et al. (2014) analyze a setting in hybrid courier operations, where dynamic pick-up requests arise throughout each period, while standard deliveries with a certain deadline are made. Another problem in courier operations is studied by Laganà et al. (2021). The dynamic multi-period general routing problem, where vertices and (un)directed edges of the considered mixed graph can both be delivery destinations, is defined in this work. The problem setting arises in combined postal service and parcel delivery systems when no estimate on the future demand is known. Finally, Subramanyam et al. (2021) study a problem setting in the delivery of industry gases, where customers may place their orders on any given day in the planning horizon and thereby reveal a demand quantity and the allowed delivery days. Only customers who have already placed their orders are known and future orders are modeled as binary random variables with known distribution. The decision problem is modelled as a multistage robust optimization problem with binary recourse decisions.

4.3.2 Literature on the PVRP

The PVRP is a generalization of the MPVRP, assuming a stationary customer demand and therefore considering a repeating planning horizon. Decisions on the delivery days are made for the long term and the defined visit schedules are assumed to repeat themselves cyclically (see Section 4.2.1).

Taxonomy for the PVRP We adapt the taxonomy developed for the MPVRP in Section 4.3.1.1 in several ways for the PVRP. First, we exclude the attribute “multiple visits” as all PVRPs consider the possibility to visit a customer multiple times (at least once in each cycle). Moreover, we introduce another classification of allowable visit periods in column “VP”. In the PVRP, there are two possible types of restrictions, either there is a required spacing between two visits or a set of visit patterns is predefined, indicated by an **S** or **P**, respectively. Again, spacing restrictions can take the flexible form, i.e., a lower and an upper bound on the spacing between two visits (e.g. Fauske et al. 2020) or the fixed form, where the required spacing is set to a fixed value (e.g., Cordeau et al. 1997). Also, the use of a predefined set of visit patterns can be further divided in separate days (e.g. Estrada-Moreno et al. 2019) and consecutive days only, which refers to an allowable visit interval (e.g., Baldacci et al. 2011). Figure 4.2 shows an example with seven periods, where in the flexible case, the spacing between visits is at least one day and at most two days and in the fixed case a visit is conducted every second day. For the predefined cases, either the separate days $\{1, 2, 4, 6\}$ can be given, or a visit interval of consecutive days $\{1, 2, 3, 4, 5\}$. The last alternative can also be used to model deadlines. Note that allowable visit periods only specify when a visit may take place and not necessarily, if a visit has to be conducted in every allowable period. This is defined together with the visiting frequency. We additionally indicate in column “VF” if the visiting frequency is a decision variable or not (e.g., Francis et al. 2006).

Reviewed literature The PVRP was introduced by Beltrami and Bodin (1974). Since then, there have been a variety of publications and different variants. Naturally, a number of surveys on the

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | ... |
|--------------------|---|---|---|---|---|---|---|---|---|-----|
| <i>Flexible</i> | | x | ← | | x | ← | x | | | |
| <i>Fixed</i> | x | ← | x | ← | x | ← | x | ← | x | |
| <i>Separate</i> | | x | x | | x | | x | | | |
| <i>Consecutive</i> | | x | x | x | x | x | | | | |

Figure 4.2: Classification of allowable visit periods.

PVRP have already been suggested. The evolution of the PVRP and some variants, namely the PVRP with time windows (PVRPTW), the PVRP with service choice and the multi-depot PVRP (MDPVRP) is studied by Francis et al. (2008). Irnich et al. (2014) propose another survey on the PVRP, with an emphasis on solution methods. The most recent comprehensive review is Campbell and Wilson (2014), who summarize research on the PVRP of forty years and reviewed the wide range of applications of the PVRP. Therefore we concentrate on works published later than 2014. Some more recent publications on the basic PVRP can also be found in Mor and Speranza (2022). However, we are not only interested in the basic version, but want to give an overview of different PVRP settings and their attributes. In total, 23 relevant publications can be found, of which 15 are motivated by substantially different real-world applications. Multiple related streams in literature can be identified. Table 4.4 shows all reviewed publications on the PVRP.

One of the most studied variants is the PVRPTW. Liu et al. (2014) study an extension of the PVRPTW that arises in the transportation of drugs and medical devices to patients' homes. Michallet et al. (2014) address an unusual PVRPTW that occurs in the field of high-value shipments, where arrival times need to be spread out within customers' time windows for a higher unpredictability to make robberies more difficult. Norouzi et al. (2015) study a PVRPTW with competition, where sales can be maximized by arriving earlier than a competitor at the customer. A variant of the PVRPTW which includes five different objectives is addressed by Wang et al. (2020), which is the only paper found to consider a type of service costs and penalty costs as well as costs dependent on the chosen visit frequency simultaneously. A container transportation problem where the planning horizon can be divided into multiple shifts is studied by Chen et al. (2020). The problem structure makes it possible to model it as a PVRPTW with open routes in which vehicles do not have to return to the depot at the end of their tours.

Another common application for the PVRP is the scheduling and routing of supply vessels for offshore installations. However, these problem settings are usually more complicated since the supply vessels make multiple-day voyages. Kisialiou et al. (2018) address a variant with flexible departure times, Borthen et al. (2018) develop a new genetic algorithm for the basic setting and outperform older solution approaches. Borthen et al. (2019) extend the problem to include a persistence objective. Recently, Vieira et al. (2021) develop a branch-and-cut algorithm as well as an adaptive large neighborhood search for these applications and achieve new best-known solutions on corresponding benchmarks.

A specific literature stream on the flexible PVRP is created by Archetti et al. (2017). They intro-

Table 4.4: Characterization of PVRPs

| Paper | Application | Obj | MP-specific | | | VRP-related ^a | | | | |
|-----------------------------|--------------|------|-------------|-------|----|--------------------------|----|----|----|----|
| | | | VC | VP | VF | HF | TW | DC | WB | MD |
| Nguyen et al. (2014) | Generic | R | | P | | x | x | x | | |
| Cacchiani et al. (2014) | Generic | R | | P | | | | | | |
| Liu et al. (2014) | Healthcare | MO | | P | | | x | | x | |
| Michallet et al. (2014) | Security | R | | P | | | x | | | |
| Ramos et al. (2014) | Waste | MO | | S | | | | x | x | x |
| Norouzi et al. (2015) | Distribution | R+O | x | P | | | x | | | |
| Lahrichi et al. (2015) | Generic | R | | P | | | | | | x |
| Gómez et al. (2015) | Waste | MO | x | S | x | x | | | x | |
| Nair et al. (2016) | Distribution | R | | P | | | | x | | |
| Archetti et al. (2017) | Generic | R | | P | | | | | | |
| Lei et al. (2017) | Surveillance | MO | x | P | | | | | | |
| Archetti et al. (2018) | Generic | R | | P | | | | | | |
| Kisialiou et al. (2018) | Maintenance | FS+R | | S | | x | x | x | x | |
| Borthen et al. (2019) | Maintenance | FS+R | | P | | x | | x | x | |
| Rothenbächer (2019) | Generic | R | | P | x | | x | | | |
| Borthen et al. (2019) | Maintenance | MO | x | P | | x | | x | x | |
| Chen et al. (2020) | Container | R | | P | | | x | | | |
| Wang et al. (2020) | Generic | MO | x | P | | | x | x | | |
| Fauske et al. (2020) | Surveillance | O | | S | x | x | | x | | |
| Frank et al. (2021) | Distribution | R | x | P | x | | | x | | |
| Vieira et al. (2021) | Maintenance | FS+R | | P | | x | | | | |
| López-Sánchez et al. (2021) | Green VRP | MO | x | P | x | | | | | |
| Huerta-Muñoz et al. (2022) | Generic | R | | P | x | x | | | | |
| Share in % | - | - | 26 | 93(P) | 26 | 35 | 35 | 39 | 26 | 9 |

^a HF: heterogeneous fleet, TW: time windows, DC: distance/duration constrained, WB: workload balancing, MD: multi-depot

duce this PVRP variant in which the number and timing of visits and the delivered quantities can be chosen freely (only subject to a maximum receiving capacity per visit) and solve small instance with branch-and-bound procedures. Archetti et al. (2018) develops a matheuristic for the problem in order to solve larger instances and Huerta-Muñoz et al. (2022) extends the problem to consider a heterogeneous vehicle fleet.

Works with generic problem settings are either devoted to introducing a new variant of the PVRP or focus on achieving better solution methods. Nguyen et al. (2014) find new best known solutions for the PVRPTW by means of a genetic algorithm with local search education operators. Cacchiani et al. (2014) are able to reach new best known solutions for the PVRP using a column generation procedure in which the sub-problem is solved by an iterated local search (ILS) heuristic. Lahrichi et al. (2015) develop a new type of solution approach which consists of multiple, parallel-running, exact or heuristic solution approaches which cooperate through an adaptive guidance mechanism, use it to solve the MDPVRP and yield new best known solutions. Rothenbächer (2019) present an exact branch-and-price-and-cut algorithm for the PVRPTW and also for a variant where they explicitly allow all kinds of schedule structures and the choice of the visit frequency.

Finally, numerous other applications show the versatility of the PVRP. A multi-objective MDPVRP arising in the collection of recyclables from drop-off containers is addressed by Ramos et al. (2014), including economic, ecological and social objectives. Nair et al. (2016) introduce a combination of

PVRP and the pickup and delivery problem with unassigned quantities of a single product that can be delivered to any customer. Lei et al. (2017) use an adapted version of the PVRP to optimize the schedule of parking enforcement patrols, where the visit at each parking lot and the routing plan of patrol vehicles are determined simultaneously. An adapted version of the PVRP is also used by Fauske et al. (2020) in order to schedule maritime surveillance activities carried out by the Norwegian military. Instead of visiting customers, maritime regions are to be scanned in certain intervals to keep information sufficiently updated. Frank et al. (2021) use a combination of the PVRP with multi-compartment vehicles to supply stores with grocery products of different segments, directly accounting for costs associated with the chosen frequency and day-combination in the objective function. Finally, López-Sánchez et al. (2021) consider a generalization of the PVRP considering the two conflicting objectives minimization of total emissions produced by all vehicles and maximization of the service frequency.

4.3.3 Solution Approaches for VRPs with Multiple Periods

We further analyze the reviewed literature streams according to the applied solution methods. Table 4.5 gives an overview of the used procedures. We introduce a separate column for a specific approach, if it is used in at least three publications. All approaches used less frequently are summarized by the columns “Others Exact” and “Others Heuristic”, respectively. Note that some papers develop multiple solution approaches, e.g., an exact procedure for smaller instances and a heuristic for larger ones.

Table 4.5: Overview of solution approaches

| Class | Exact approaches ^a | | | | Heuristic approaches ^b | | | | | | | |
|--------|-------------------------------|------------|-----------------|----------------|-----------------------------------|-----|-----|----|-----|------|---------------------|--------------------|
| | B&C | B&P /CG | Others Exact | Exact Total | TS | LNS | VNS | GA | ILS | Math | Others Heuristic | Heuristic Total |
| PVRP | 3 | 1 | 1 | 5 | 2 | 2 | 1 | 4 | 2 | 5 | 4 | 20 |
| MPVRP | 5 | 2 | 0 | 7 | 2 | 2 | 1 | 0 | 1 | 0 | 2 | 8 |
| DMPVRP | 2 | 3 | 0 | 5 | 0 | 1 | 3 | 2 | 1 | 0 | 6 | 13 |
| Sum | 10 | 6 | 1 | 17 | 4 | 5 | 5 | 6 | 4 | 5 | 12 | 41 |

^a B&C: branch-and-cut, B&P/CG: branch-and-price/column generation

^b TS: tabu search, LNS: large neighborhood search, VNS: variable neighborhood search, GA: genetic algorithm, ILS: iterated local search, Math: matheuristic

One can observe that many papers use one of two extremes: branch-and-cut procedures on the one side which mostly refer to the use of common solvers, and specialized, self-developed heuristic approaches on the other side. In summary however, a variety of solution methods have already been applied to all three problem classes, without a single approach dominating the others.

4.4 Conclusion

In this paper we provide a comprehensive view on VRPs with multiple periods, namely the PVRP and the MPVRP. We examine the inconsistent use of the terms PVRP and MPVRP and propose the criterion of periodicity to distinguish between the two problem variants, as many publications

already do. We formulate a general MPVRP as a basis for following discussions that is easy to complement with typical constraints arising in the field of multi-period tour planning. We structure our comprehensive literature review according to a typology based on attributes characteristic to the feature of multiple periods.

The literature on the MPVRP can be subdivided in static and dynamic problem settings. Specifically the DMPVRP has recently been studied extensively. It is to be expected that especially the combination of dynamic MPVRPs with stochastic elements will continue to see much research, as the large amounts of data often available today enable decision makers to predict information that was previously only revealed over time. For the PVRP, numerous recent publications can be found, arising in a broad range of applications, showing the versatility of this problem. Since the last survey by Campbell and Wilson (2014), particularly the extension of with time windows and the incorporation of multiple objectives have gained importance. As a future research avenue, the areas of other related problem classes where multi-period settings arise, but are not a core feature nor of special interest like the consistent VRP (e.g. Kovacs et al. 2014), the rich VRP (e.g. Lahyani et al. 2015) or the multi-compartment VRP (e.g. Ostermeier et al. 2021) are to be examined for the influence of the multi-period feature.

Bibliography

- Agatz, N., Campbell, A., Fleischmann, M., Savelsbergh, M., 2011. Time slot management in attended home delivery. *Transportation Science* 45 (3), 435–449.
- Albareda-Sambola, M., Fernández, E., Laporte, G., 2014. The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers & Operations Research* 48, 31–39.
- Amarouche, Y., Guibadj, R. N., Moukrim, A., 2018. A neighborhood search and set cover hybrid heuristic for the two-echelon vehicle routing problem. In: Borndörfer, R., Storandt, S. (Eds.), 18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018). Vol. 65 of OpenAccess Series in Informatics (OASICs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 11:1–11:15.
URL <http://drops.dagstuhl.de/opus/volltexte/2018/9716>
- Angelelli, E., Bianchessi, N., Mansini, R., Speranza, M. G., 2009a. Management policies in a dynamic multi period routing problem. In: *Innovations in distribution logistics*. Springer, pp. 1–15.
- Angelelli, E., Bianchessi, N., Mansini, R., Speranza, M. G., 2009b. Short term strategies for a dynamic multi-period routing problem. *Transportation Research Part C: Emerging Technologies* 17 (2), 106–119.
- Angelelli, E., Bianchessi, N., Mansini, R., Speranza, M. G., 2010. Comparison of policies in dynamic routing problems. *Journal of the Operational Research Society* 61 (4), 686–695.
- Angelelli, E., Grazia Speranza, M., Savelsbergh, M. W., 2007a. Competitive analysis for dynamic multiperiod uncapacitated routing problems. *Networks: An International Journal* 49 (4), 308–317.
- Angelelli, E., Savelsbergh, M. W., Speranza, M. G., 2007b. Competitive analysis of a dispatch policy for a dynamic multi-period routing problem. *Operations Research Letters* 35 (6), 713–721.
- Archetti, C., Fernández, E., Huerta-Muñoz, D. L., 2017. The flexible periodic vehicle routing problem. *Computers & Operations Research* 85, 58 – 70.
- Archetti, C., Fernández, E., Huerta-Muñoz, D. L., 2018. A two-phase solution algorithm for the flexible periodic vehicle routing problem. *Computers & Operations Research* 99, 27 – 37.
- Archetti, C., Jabali, O., Speranza, M. G., 2015. Multi-period vehicle routing problem with due dates. *Computers & Operations Research* 61, 122 – 134.
- Arnold, F., Sörensen, K., 2019. Knowledge-guided local search for the vehicle routing problem. *Computers & Operations Research* 105, 32–46.
- Arnold, F., Sörensen, K., 2021. A progressive filtering heuristic for the location-routing problem and variants. *Computers & Operations Research* 129, 105166.
- Athanasopoulos, T., Minis, I., 2013. Efficient techniques for the multi-period vehicle routing problem with time windows within a branch and price framework. *Annals of Operations Research* 206 (1), 1 – 22.
- Avella, P., Boccia, M., Sforza, A., 2004. Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research* 152 (1), 170–179.
- Baldacci, R., Bartolini, E., Mingozzi, A., Valletta, A., 2011. An exact algorithm for the period routing problem. *Operations Research* 59 (1), 228–241.

- Barreto, S., Ferreira, C., Paixão, J., Santos, B. S., 2007. Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research* 179 (3), 968–977.
- Belhaiza, S., Hansen, P., Laporte, G., 2014. A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Computers & Operations Research* 52, 269–281.
- Belhaiza, S., M’Hallah, R., Ben Brahim, G., 2017. A new hybrid genetic variable neighborhood search heuristic for the vehicle routing problem with multiple time windows. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 1319–1326.
- Belhaiza, S., M’Hallah, R., Brahim, G. B., Laporte, G., 2019. Three multi-start data-driven evolutionary heuristics for the vehicle routing problem with multiple time windows. *Journal of Heuristics* 25 (3), 485–515.
- Beltrami, E. J., Bodin, L. D., 1974. Networks and vehicle routing for municipal waste collection. *Networks* 4 (1), 65–94.
- Ben-Ameur, W., 2004. Computing the initial temperature of simulated annealing. *Computational Optimization and Applications* 29 (3), 369–385.
- Billing, C., Jaehn, F., Wensing, T., 2018. A multiperiod auto-carrier transportation problem with probabilistic future demands. *Journal of Business Economics* 88 (7), 1009–1028.
- Bonassa, A. C., Cunha, C. B., Isler, C. A., 2019. An exact formulation for the multi-period auto-carrier loading and transportation problem in brazil. *Computers & Industrial Engineering* 129, 144–155.
- Borthen, T., Loennechen, H., Fagerholt, K., Wang, X., Vidal, T., 2019. Bi-objective offshore supply vessel planning with costs and persistence objectives. *Computers & Operations Research* 111, 285 – 296.
- Borthen, T., Loennechen, H., Wang, X., Fagerholt, K., Vidal, T., 2018. A genetic search-based heuristic for a fleet size and periodic routing problem with application to offshore supply planning. *EURO Journal on Transportation and Logistics* 7 (2), 121–150.
- Bostel, N., Dejax, P., Guez, P., Tricoire, F., 2008. Multiperiod planning and routing on a rolling horizon for field force optimization logistics. In: *The vehicle routing problem: latest advances and new challenges*. Springer, pp. 503–525.
- Breunig, U., Schmid, V., Hartl, R. F., Vidal, T., 2016. A large neighbourhood based heuristic for two-echelon routing problems. *Computers & Operations Research* 76, 208–225.
- Broekmeulen, R., van Donselaar, K., Fransoo, J., Woensel, T. v., 2006. The opportunity of excess shelf space in grocery retail store. *Operations Research* 49, 710–719.
- Cacchiani, V., Hemmelmayr, V. C., Tricoire, F., 2014. A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics* 163, 53–64.
- Campbell, A. M., Wilson, J., 2014. Forty years of periodic vehicle routing. *Networks* 63 (1), 2–15.
- Cantu-Funes, R., Salazar-Aguilar, M. A., Boyer, V., 2018. Multi-depot periodic vehicle routing problem with due dates and time windows. *Journal of the Operational Research Society* 69 (2), 296 – 306.
- Cardenas, I., Beckers, J., Vanelslander, T., Verhetsel, A., Dewulf, W., 2016. Spatial characteristics of failed and successful e-commerce deliveries in belgian cities. In: *Information Systems, Logistic and Supply Chain Conf 2016*.
- Chao, I.-M., Golden, B. L., Wasil, E., 1993. A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. *American Journal of Mathematical and Management Sciences* 13 (3-4), 371–406.
- Chen, B., Qu, R., Bai, R., Laesanklang, W., 2020. A variable neighborhood search algorithm with reinforcement learning for a real-life periodic vehicle routing problem with time windows and open routes. *RAIRO – Operations Research* 54 (5), 1467 – 1494.

Bibliography

- Christofides, N., Beasley, J. E., 1984. The period routing problem. *Networks* 14 (2), 237–256.
- Clarke, G., Wright, J. W., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12 (4), 568–581.
- Coelho, L. C., Laporte, G., 2015. Classification, models and exact algorithms for multi-compartment delivery problems. *European Journal of Operational Research* 242 (3), 854–864.
- Contardo, C., Cordeau, J.-F., Gendron, B., 2013. A GRASP + ILP-based metaheuristic for the capacitated location-routing problem. *Journal of Heuristics* 20 (1), 1–38.
- Contardo, C., Cordeau, J.-F., Gendron, B., 2014. An exact algorithm based on cut-and-column generation for the capacitated location-routing problem. *INFORMS Journal on Computing* 26 (1), 88–102.
- Cordeau, J.-F., Dell’Amico, M., Falavigna, S., Iori, M., 2015. A rolling horizon algorithm for auto-carrier transportation. *Transportation Research Part B: Methodological* 76, 68–80.
- Cordeau, J.-F., Gendreau, M., Laporte, G., 1997. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks: An International Journal* 30 (2), 105–119.
- Cordeau, J.-F., Maischberger, M., 2012. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research* 39 (9), 2033–2050.
- Crainic, T. G., Mancini, S., Perboli, G., Tadei, R., 2011. Multi-start heuristics for the two-echelon vehicle routing problem. In: *Evolutionary Computation in Combinatorial Optimization*. Springer Berlin Heidelberg, pp. 179–190.
- Curşeu, A., van Woensel, T., Fransoo, J., van Donselaar, K., Broekmeulen, R., 2009. Modelling handling operations in grocery retail stores: An empirical analysis. *Journal of the Operational Research Society* 60 (2), 200–214.
- Darvish, M., Coelho, L. C., Laporte, G., 2020. The Dynamic Routing Problem with Due Dates and Stochastic Release Dates. *Faculté des sciences de l’administration*.
- Dayarian, I., Crainic, T. G., Gendreau, M., Rei, W., 2015. A branch-and-price approach for a multi-period vehicle routing problem. *Computers & Operations Research* 55, 167 – 184.
- Dayarian, I., Crainic, T. G., Gendreau, M., Rei, W., 2016. An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research: Part E* 95, 95 – 123.
- de Jong, C., Kant, G., van Vliet, A., 1996. On finding minimal route duration in the vehicle routing problem with multiple time windows. Manuscript, Department of Computer Science, Utrecht University, Holland.
- 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.
- Donselaar, K. H. v., Gaur, V., Woensel, T. v., Broekmeulen, R. A., Fransoo, J. C., 2010. Ordering behavior in retail stores and implications for automated replenishment. *Management Science* 56 (5), 766–784.
- Drexl, M., Schneider, M., 2015. A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research* 241 (2), 283–308.
- Dueck, G., 1993. New optimization heuristics. *Journal of Computational Physics* 104 (1), 86–92.
- Duhamel, C., Lacomme, P., Prins, C., Prodhon, C., 2010. A GRASP×ELS approach for the capacitated location-routing problem. *Computers & Operations Research* 37 (11), 1912–1923.
- Edwards, J., McKinnon, A., Cherrett, T., McLeod, F., Song, L., 2009. The impact of failed home deliveries on carbon emissions: Are collection/delivery points environmentally-friendly alternatives. In: *14th Annual Logistics Research Network Conference*. Citeseer, p. M117.

Bibliography

- Enthoven, D. L., Jargalsaikhan, B., Roodbergen, K. J., uit het Broek, M. A., Schrotenboer, A. H., 2020. The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. *Computers & Operations Research* 118, 104919.
- Escobar, J. W., Linfati, R., Baldoquin, M. G., Toth, P., 2014a. A granular variable tabu neighborhood search for the capacitated location-routing problem. *Transportation Research Part B: Methodological* 67, 344–356.
- Escobar, J. W., Linfati, R., Toth, P., 2013. A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Computers & Operations Research* 40 (1), 70–79.
- Escobar, J. W., Linfati, R., Toth, P., Baldoquin, M. G., 2014b. A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. *Journal of Heuristics* 20 (5), 483–509.
- Estrada-Moreno, A., Savelsbergh, M., Juan, A. A., Panadero, J., 2019. Biased-randomized iterated local search for a multiperiod vehicle routing problem with price discounts for delivery flexibility. *International Transactions in Operational Research* 26 (4), 1293–1314.
- Fauske, M. F., Mannino, C., Ventura, P., 2020. Generalized periodic vehicle routing and maritime surveillance. *Transportation Science* 54 (1), 164 – 183.
- Favaretto, D., Moretti, E., Pellegrini, P., 2007. Ant colony system for a VRP with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics* 10 (2), 263–284.
- Figliozzi, M. A., 2010. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C: Emerging Technologies* 18 (5), 668–679.
- Fisher, M. L., Jaikumar, R., 1981. A generalized assignment heuristic for vehicle routing. *Networks* 11 (2), 109–124.
- Florio, A. M., Feillet, D., Hartl, R. F., 2018. The delivery problem: Optimizing hit rates in e-commerce deliveries. *Transportation Research Part B: Methodological* 117, 455–472.
- Francis, P., Smilowitz, K., Tzur, M., 2006. The period vehicle routing problem with service choice. *Transportation science* 40 (4), 439–454.
- Francis, P. M., Smilowitz, K. R., Tzur, M., 2008. The period vehicle routing problem and its extensions. In: *The vehicle routing problem: latest advances and new challenges*. Springer, pp. 73–102.
- Frank, M., Ostermeier, M., Holzapfel, A., Hübner, A., Kuhn, H., 2021. Optimizing routing and delivery patterns with multi-compartment vehicles. *European Journal of Operational Research* 293 (2), 495 – 510.
- Fu, Z., Eglese, R., Li, L. Y. O., 2008. A unified tabu search algorithm for vehicle routing problems with soft time windows. *Journal of the Operational Research Society* 59 (5), 663–673.
- Gaudioso, M., Paletta, G., 1992. A heuristic for the periodic vehicle routing problem. *Transportation Science* 26 (2), 86–92.
- Gaur, V., Fisher, M., 2004. A periodic inventory routing problem at a supermarket chain. *Operations Research* 52, 813–822.
- Gershuny, J., Sullivan, O., 2017. *United Kingdom Time Use Survey, 2014-2015*.
- Glatzel, C., Großpietsch, J., Hübner, A., 2012. Higher margins through efficient supply chains. *Akzente* (2), 16–21.
- Gómez, J. R., Pacheco, J., Gonzalo-Orden, H., 2015. A tabu search method for a bi-objective urban waste collection problem. *Computer-Aided Civil and Infrastructure Engineering* 30 (1), 36–53.

Bibliography

- Hashimoto, H., Ibaraki, T., Imahori, S., Yagiura, M., 2006. The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics* 154 (16), 2271–2290.
- Hemmelmayr, V. C., Cordeau, J.-F., Crainic, T. G., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research* 39 (12), 3215–3228.
- Henke, T., Speranza, M. G., Wäscher, G., 2015. The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research* 246 (3), 730–743.
- Henke, T., Speranza, M. G., Wäscher, G., 2019. A branch-and-cut algorithm for the multi-compartment vehicle routing problem with flexible compartment sizes. *Annals of Operations Research* 275 (2), 321–338.
- Hermes, 2020. Allgemeine Geschäftsbedingungen für den Versand von Hermes Päckchen. Hermes Germany GmbH.
URL https://www.myhermes.de/content/pdf/agb_verpackungsrichtlinien.pdf
- Holzapfel, A., Hübner, A., Kuhn, H., Sternbeck, M., 2016. Delivery pattern and transportation planning in grocery retailing. *European Journal of Operational Research* 252 (1), 54–68.
- Hoogeboom, M., Dullaert, W., Lai, D., Vigo, D., 2020. Efficient neighborhood evaluations for the vehicle routing problem with multiple time windows. *Transportation Science* 54 (2), 400–416.
- Hsiao, Y.-H., Chen, M.-C., Chin, C.-L., 2017. Distribution planning for perishable foods in cold chains with quality concerns: Formulation and solution procedure. *Trends in Food Science & Technology* 61, 80–93.
- Hübner, A., Ostermeier, M., 2019. A multi-compartment vehicle routing problem with loading and unloading costs. *Transportation Science* 53 (1), 282–300.
- Hübner, A. H., Kuhn, H., Sternbeck, M. G., 2013. Demand and supply chain planning in grocery retail: An operations planning framework. *International Journal of Retail & Distribution Management* 41 (7), 512–530.
- Huerta-Muñoz, D. L., Archetti, C., Fernández, E., Perea, F., 2022. The heterogeneous flexible periodic vehicle routing problem: Mathematical formulations and solution algorithms. *Computers & Operations Research* 141, N.PAG.
- Hübner, A., Kuhn, H., Wollenburg, J., 2016. Last mile fulfilment and distribution in omni-channel grocery retailing. *International Journal of Retail & Distribution Management* 44 (3), 228–247.
- Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., Yagiura, M., 2005. Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation Science* 39 (2), 206–232.
- Ibaraki, T., Imahori, S., Nonobe, K., Sobue, K., Uno, T., Yagiura, M., 2008. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics* 156 (11), 2050–2069.
- Irnich, S., Schneider, M., Vigo, D., 2014. Chapter 9: Four variants of the vehicle routing problem. In: *Vehicle Routing: Problems, Methods, and Applications*, Second Edition. SIAM, pp. 241–271.
- Jie, W., Yang, J., Zhang, M., Huang, Y., 2019. The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. *European Journal of Operational Research* 272 (3), 879–904.
- Johnson, D. S., Aragon, C. R., McGeoch, L. A., Schevon, C., 1989. Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations Research* 37 (6), 865–892.
- Kim, J.-S., Lee, D.-H., 2015. An integrated approach for collection network design, capacity planning and vehicle routing in reverse logistics. *Journal of the Operational Research Society* 66 (1), 76 – 85.

Bibliography

- Kisialiou, Y., Gribkovskaia, I., Laporte, G., 2018. The periodic supply vessel planning problem with flexible departure times and coupled vessels. *Computers & Operations Research* 94, 52 – 64.
- 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, Otto-von-Guericke-University Magdeburg.
- Koskosidis, Y. A., Powell, W. B., Solomon, M. M., 1992. An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation Science* 26 (2), 69–85.
- Kotzab, H., Teller, C., 2005. Development and empirical test of a grocery retail instore logistics model. *British Food Journal* 107, 594–605.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., Parragh, S. N., 2014. Vehicle routing problems in which consistency considerations are important: A survey. *Networks* 64 (3), 192–213.
- Kritzing, S., Tricoire, F., Doerner, K. F., Hartl, R. F., Stützle, T., 2017. A unified framework for routing problems with a fixed fleet size. *International Journal of Metaheuristics* 6 (3), 160–209.
- Kuhn, H., Sternbeck, M. G., 2013. Integrative retail logistics: An exploratory study. *Operations Management Research* 6 (1-2), 2–18.
- Laganà, D., Laporte, G., Vocaturo, F., 2021. A dynamic multi-period general routing problem arising in postal service and parcel delivery systems. *Computers & Operations Research* 129, N.PAG.
- Lahrichi, N., Crainic, T. G., Gendreau, M., Rei, W., Crişan, G. C., Vidal, T., 2015. An integrative cooperative search framework for multi-decision-attribute combinatorial optimization: Application to the mdprp. *European Journal of Operational Research* 246 (2), 400 – 412.
- Lahyani, R., Khemakhem, M., Semet, F., 2015. Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research* 241 (1), 1–14.
- Laporte, G., 2009. Fifty years of vehicle routing. *Transportation Science* 43 (4), 408–416.
- Larrain, H., Coelho, L. C., Archetti, C., Speranza, M. G., 2019. Exact solution methods for the multi-period vehicle routing problem with due dates. *Computers & Operations Research* 110, 148 – 158.
- Larsen, R., Pacino, D., 2019. Fast delta evaluation for the vehicle routing problem with multiple time windows.
URL <https://arxiv.org/abs/1905.04114>
- Lei, C., Zhang, Q., Ouyang, Y., 2017. Planning of parking enforcement patrol considering drivers' parking payment behavior. *Transportation Research: Part B* 106, 375 – 392.
- Liu, R., Xie, X., Garaix, T., 2014. Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics. *Omega* 47, 17 – 32.
- Lopes, R. B., Ferreira, C., Santos, B. S., 2016. A simple and effective evolutionary algorithm for the capacitated location–routing problem. *Computers & Operations Research* 70, 155–162.
- Luo, J., Chen, M.-R., 2014. Multi-phase modified shuffled frog leaping algorithm with extremal optimization for the MDVRP and the MDVRPTW. *Computers & Industrial Engineering* 72, 84–97.
- López-Sánchez, A. D., Molina, J., Laguna, M., Hernández-Díaz, A. G., 2021. Optimizing a bi-objective vehicle routing problem that appears in industrial enterprises. *Expert Systems* 38 (1), 1 – 18.
- Marques, G., Sadykov, R., Deschamps, J.-C., Dupas, R., 2020. An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Computers & Operations Research* 114, 104833.
- Martins, S., Ostermeier, M., Amorim, P., Hübner, A., Almada-Lobo, B., 2019. Product-oriented time window assignment for a multi-compartment vehicle routing problem. *European Journal of Operational Research* 276, 893–909.

Bibliography

- Michallet, J., Prins, C., Amodeo, L., Yalaoui, F., Vitry, G., 2014. Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers & Operations Research* 41, 196 – 207.
- Minner, S., Transchel, S., 2010. Periodic review inventory-control for perishable products under service-level constraints. *OR Spectrum* 32 (4), 979–996.
- Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., Herazo-Padilla, N., 2015. A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering* 79, 115–129.
- Mor, A., Speranza, M. G., 2022. Vehicle routing problems over time: a survey. *Annals of Operations Research*, 1–21.
- Mourgaya, M., Vanderbeck, F., 2006. The periodic vehicle routing problem: classification and heuristic. *RAIRO-Operations Research* 40 (2), 169–194.
- Mouthuy, S., Massen, F., Deville, Y., Van Hentenryck, P., 2015. A multistage very large-scale neighborhood search for the vehicle routing problem with soft time windows. *Transportation Science* 49 (2), 223–238.
- Muñoz-Villamizar, A., Velázquez-Martínez, J. C., Mejía-Argueta, C., Gámez-Pérez, K., 2021. The impact of shipment consolidation strategies for green home delivery: a case study in a mexican retail company. *International Journal of Production Research*, 1–18.
- Mühlbauer, F., Fontaine, P., 2021. A parallelised large neighbourhood search heuristic for the asymmetric two-echelon vehicle routing problem with swap containers for cargo-bicycles. *European Journal of Operational Research* 289 (2), 742 – 757.
- Nagata, Y., 1997. Edge assembly crossover: A high-power genetic algorithm for the traveling salesman problem. In: *Proc. of 7th Int. Conf. on Genetic Algorithms*, 1997.
- Nagata, Y., Bräysy, O., Dullaert, W., 2010. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research* 37 (4), 724–737.
- Nair, D. J., Grzybowska, H., Rey, D., Dixit, V., 2016. Food rescue and delivery: Heuristic algorithm for periodic unpaired pickup and delivery vehicle routing problem. *Transportation Research Record* 2548 (1), 81–89.
- Nguyen, P. K., Crainic, T. G., Toulouse, M., 2014. A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows. *Journal of Heuristics* 20 (4), 383–416.
- Ninikas, G., Athanasopoulos, T., Zempeki, V., Minis, I., 2014. Integrated planning in hybrid courier operations. *International Journal of Logistics Management* 25 (3), 611 – 634.
- Norouzi, N., Sadegh-Amalnick, M., Alinaghiyan, M., 2015. Evaluating of the particle swarm optimization in a periodic vehicle routing problem. *Measurement* 62, 162–169.
- Okholm, H. B., Thelle, M. H., Möller, A., Basalisco, B., Rolmer, S., 2013. E-commerce and delivery: A study of the state of play of eu parcel markets with particular emphasis on e-commerce. *European Commission*.
- Ostermeier, M., Henke, T., Hübner, A., Wäscher, G., 2021. Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. *European Journal of Operational Research* 292 (3), 799–817.
- Ostermeier, M., Hübner, A., 2018. Vehicle selection for a multi-compartment vehicle routing problem. *European Journal of Operational Research* 269, 682–694.
- Ostermeier, M., Martins, S., Amorim, P., Hübner, A., 2018. Loading constraints for a multi-compartment vehicle routing problem. *OR Spectrum* 40, 997 – 1027.

Bibliography

- Özarık, S. S., Veelenturf, L. P., Van Woensel, T., Laporte, G., 2021. Optimizing e-commerce last-mile vehicle routing and scheduling under uncertain customer presence. *Transportation Research Part E: Logistics and Transportation Review* 148, 102263.
- Pacheco, J., Alvarez, A., Garcia, I., Angel-Bello, F., 2012. Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates. *Journal of the Operational Research Society* 63 (5), 569–581.
- Pan, S., Giannikas, V., Han, Y., Grover-Silva, E., Qiao, B., 2017. Using customer-related data to enhance e-grocery home delivery. *Industrial Management & Data Systems* 117 (9), 1917–1933.
- Pasha, U., Hoff, A., Hvattum, L. M., 2016. Simple heuristics for the multi-period fleet size and mix vehicle routing problem. *INFOR* 54 (2), 97 – 120.
- Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science* 45 (3), 364–380.
- Pirkwieser, S., Raidl, G. R., 2010. Variable neighborhood search coupled with ILP-based very large neighborhood searches for the (periodic) location-routing problem. In: *Hybrid Metaheuristics*. Springer Berlin Heidelberg, pp. 174–189.
- Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research* 34 (8), 2403–2435.
- Pitney Bowes, 2021. Pitney Bowes shipping index 2021.
URL https://www.pitneybowes.com/content/dam/pitneybowes/us/en/shipping-index/parcel_shipping_index_ebook_final.pdf
- Praet, S., Martens, D., 2019. Efficient parcel delivery by predicting customers’ locations. *Decision Sciences* 51 (5), 1202–1231.
- Prins, C., Prodhon, C., Calvo, R. W., 2004. Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité. *MOSIM’04* 2, 1115–1122.
- Prins, C., Prodhon, C., Calvo, R. W., 2006a. A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. In: *Evolutionary Computation in Combinatorial Optimization*. Springer Berlin Heidelberg, pp. 183–194.
- Prins, C., Prodhon, C., Calvo, R. W., 2006b. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR* 4 (3), 221–238.
- Prins, C., Prodhon, C., Ruiz, A., Soriano, P., Calvo, R. W., 2007. Solving the capacitated location-routing problem by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation Science* 41 (4), 470–483.
- Prodhon, C., Prins, C., 2008. A memetic algorithm with population management (MA|PM) for the periodic location-routing problem. In: *Hybrid Metaheuristics*. Springer Berlin Heidelberg, pp. 43–57.
- Prodhon, C., Prins, C., 2014. A survey of recent research on location-routing problems. *European Journal of Operational Research* 238 (1), 1–17.
- Psaraftis, H. N., Wen, M., Kontovas, C. A., 2016. Dynamic vehicle routing problems: Three decades and counting. *Networks* 67 (1), 3–31.
- Punakivi, M., Saranen, J., 2001. Identifying the success factors in e-grocery home delivery. *International Journal of Retail & Distribution Management* 29 (4), 156–163.
- Punakivi, M., Yrjölä, H., Holmström, J., 2001. Solving the last mile issue: Reception box or delivery box? *International Journal of Physical Distribution & Logistics Management* 31 (6), 427–439.
- 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.

Bibliography

- Ramos, T. R. P., Gomes, M. I., Barbosa-Póvoa, A. P., 2014. Planning a sustainable reverse logistics system: Balancing costs with environmental and social concerns. *Omega* 48, 60 – 74.
- Reiner, G., Teller, C., Kotzab, H., 2013. Analyzing the efficient execution on in-store logistics processes in grocery retailing – the case of dairy products. *Production and Operations Management* 22 (4), 924–939.
- Renaud, J., Laporte, G., Boctor, F. F., 1996. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research* 23 (3), 229–235.
- Ritzinger, U., Puchinger, J., Hartl, R. F., 2016. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research* 54 (1), 215–231.
- Ronen, D., Goodhart, C. A., 2008. Tactical store delivery planning. *Journal of the Operational Research Society* 59 (8), 1047–1054.
- Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 40 (4), 455–472.
- Rothenbächer, A.-K., 2019. Branch-and-price-and-cut for the periodic vehicle routing problem with flexible schedule structures. *Transportation Science* 53 (3), 850 – 866.
- Russell, R., Igo, W., 1979. An assignment routing problem. *Networks* 9 (1), 1–17.
- Sadati, M. E. H., Çatay, B., Aksen, D., 2021. An efficient variable neighborhood search with tabu shaking for a class of multi-depot vehicle routing problems. *Computers & Operations Research* 133, 105269.
- Sadykov, R., Uchoa, E., Pessoa, A., 2021. A bucket graph-based labeling algorithm with application to vehicle routing. *Transportation Science* 55 (1), 4–28.
- Schaap, H., Schiffer, M., Schneider, M., Walther, G., 2019. A large neighborhood search for the vehicle routing problem with multiple time windows. Working Paper.
- Schneider, M., Drexl, M., 2017. A survey of the standard location-routing problem. *Annals of Operations Research* 259 (1-2), 389–414.
- Schneider, M., Löffler, M., 2019. Large composite neighborhoods for the capacitated location-routing problem. *Transportation Science* 53 (1), 301–318.
- Schönberger, J., 2016. Multi-period vehicle routing with limited period load. *IFAC-PapersOnLine* 49 (2), 24–29.
- Schrotenboer, A. H., uit het Broek, M. A., Jargalsaikhan, B., Roodbergen, K. J., 2018. Coordinating technician allocation and maintenance routing for offshore wind farms. *Computers & Operations Research* 98, 185–197.
- Schwerdfeger, S., Boysen, N., 2020. Optimizing the changing locations of mobile parcel lockers in last-mile distribution. *European Journal of Operational Research* 285 (3), 1077–1094.
- Shaw, P., 1997. A new local search algorithm providing high quality solutions to vehicle routing problems. APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK.
- Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., Woensel, T. V., 2022. Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*.
- Soeffker, N., Ulmer, M. W., Mattfeld, D. C., 2021. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research*.
- Solomon, M. M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35 (2), 254–265.
- Solomon, M. M., Desrosiers, J., 1988. Survey paper—time window constrained routing and scheduling problems. *Transportation Science* 22 (1), 1–13.

Bibliography

- Sternbeck, M. G., Kuhn, H., 2014. An integrative approach to determine store delivery patterns in grocery retailing. *Transportation Research Part E* 70, 205–224.
- Subramanian, A., Uchoa, E., Ochi, L. S., 2013. A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research* 40 (10), 2519–2531.
- Subramanyam, A., Mufalli, F., Laínez-Aguirre, J. M., Pinto, J. M., Gounaris, C. E., 2021. Robust multiperiod vehicle routing under customer order uncertainty. *Operations Research* 69 (1), 30–60.
- Taillard, É., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.-Y., 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* 31 (2), 170–186.
- Taube, F., Minner, S., 2018. Data-driven assignment of delivery patterns with handling effort considerations in retail. *Computers & Operations Research* 100, 379–393.
- Ting, C.-J., Chen, C.-H., 2013. A multiple ant colony optimization algorithm for the capacitated location routing problem. *International Journal of Production Economics* 141 (1), 34–44.
- Toth, P., Vigo, D., 2014. *Vehicle Routing: Problems, Methods, and Applications*, Second Edition. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics.
- Tricoire, F., 2007. Vehicle and personnel routing optimization in the service sector: application to water distribution and treatment. Ph.D. thesis, Springer.
- Tuzun, D., Burke, L. I., 1999. A two-phase tabu search approach to the location routing problem. *European Journal of Operational Research* 116 (1), 87–99.
- Ulmer, M. W., 2020. Horizontal combinations of online and offline approximate dynamic programming for stochastic dynamic vehicle routing. *Central European Journal of Operations Research* 28 (1), 279–308.
- Ulmer, M. W., Soeffker, N., Mattfeld, D. C., 2018. Value function approximation for dynamic multi-period vehicle routing. *European Journal of Operational Research* 269 (3), 883–899.
- van Duin, J. H. R., de Goffau, W., Wiegmans, B., Tavasszy, L. A., Saes, M., 2016. Improving home delivery efficiency by using principles of address intelligence for B2C deliveries. *Transportation Research Procedia* 12, 14–25.
- van Zelst, S., van Donselaar, K., van Woensel, T., Broekmeulen, R., Fransoo, J., 2009. Logistics drivers for shelf stacking in grocery retail stores: Potential for efficiency improvement. *International Journal of Production Economics* 121 (2), 620–632.
- Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., Rei, W., 2012. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research* 60 (3), 611–624.
- Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., 2013a. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research* 231 (1), 1–21.
- Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., 2013b. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research* 40 (1), 475–489.
- Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., 2014a. Implicit depot assignments and rotations in vehicle routing heuristics. *European Journal of Operational Research* 237 (1), 15–28.
- Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., 2014b. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research* 234 (3), 658–673.
- Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., 2015. Timing problems and algorithms: Time decisions for sequences of activities. *Networks* 65 (2), 102–128.
- Vidal, T., Laporte, G., Matl, P., 2020. A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research* 286 (2), 401–416.

Bibliography

- Vieira, B. S., Ribeiro, G. M., Bahiense, L., Cruz, R., Mendes, A. B., Laporte, G., 2021. Exact and heuristic algorithms for the fleet composition and periodic routing problem of offshore supply vessels with berth allocation decisions. *European Journal of Operational Research* 295 (3), 908 – 923.
- Voigt, S., Frank, M., Fontaine, P., Kuhn, H., 2021. The vehicle routing problem with availability profiles. Working Paper.
- Voigt, S., Kuhn, H., 2021. Crowdsourced logistics: The pickup and delivery problem with transshipments and occasional drivers. *Networks*.
- von Abrams, K., 2021. Global ecommerce forecast 2021.
URL <https://www.emarketer.com/content/global-ecommerce-forecast-2021>
- Wang, J., Ren, W., Zhang, Z., Huang, H., Zhou, Y., 2020. A hybrid multiobjective memetic algorithm for multiobjective periodic vehicle routing problem with time windows. *IEEE Transactions on Systems, Man & Cybernetics. Systems* 50 (11), 4732 – 4745.
- Wang, K., Shao, Y., Zhou, W., 2017. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transportation Research Part D: Transport and Environment* 57, 262–276.
- Wen, M., Cordeau, J.-F., Laporte, G., Larsen, J., 2010. The dynamic multi-period vehicle routing problem. *Computers & Operations Research* 37 (9), 1615 – 1623.
- Wong, R. T., 2008. Vehicle routing for small package delivery and pickup services. In: *Operations Research/Computer Science Interfaces*. Springer US, pp. 475–485.
- Yu, B., Yang, Z.-Z., Xie, J.-X., 2011. A parallel improved ant colony optimization for multi-depot vehicle routing problem. *Journal of the Operational Research Society* 62 (1), 183–188.
- Yu, V. F., Jodiawan, P., Hou, M.-L., Gunawan, A., 2021. Design of a two-echelon freight distribution system in last-mile logistics considering covering locations and occasional drivers. *Transportation Research Part E: Logistics and Transportation Review* 154, 102461.
- Yu, V. F., Lin, S.-W., Lee, W., Ting, C.-J., 2010. A simulated annealing heuristic for the capacitated location routing problem. *Computers & Industrial Engineering* 58 (2), 288–299.
- Zajac, S., 2017. An adaptive large neighborhood search for the periodic vehicle routing problem. In: Bektaş, T., Coniglio, S., Martinez-Sykora, A., Voß, S. (Eds.), *Computational Logistics*. Springer International Publishing, Cham, pp. 34–48.

Appendix A

Appendix for The Vehicle Routing Problem with Availability Profiles

A.1 Parameters

Table A.1: Parameters for HALNS

| Parameter | Meaning | Chosen Value |
|---------------------|---|--|
| n_p | Size of the initial population | 48 |
| gen^{\max} | Number of generations | 30 |
| gen^{stop} | Number of generations without improvement | 10 |
| it^{limit} | Number of iterations without improvement (one ALNS run) | 30,000 (initial population) 7,500 (following generations) |
| β | Cool rate in SA | 0.9999 |
| χ_0 | Acceptance probability in SA | 0.25 |
| n_0 | Number of iterations for determining the initial SA temperature | 400 |
| ω | Weight for penalties drawn for every ALNS run | $unif(0.01, 1.00) \cdot C^{\text{VRP}}$ |
| p^{binom} | Probability for binomial distribution drawn for every ALNS run | $unif(0.12, 0.24)$ |
| σ_1 | Score for operator - new best solution | 35 |
| σ_2 | Score for operator - new best current solution | 2 |
| σ_3 | Score for operator - worse solution, but accepted via SA | 1 |

A.2 Results for VRPTW, VRPMTW and VRPSTW Benchmark Instances

Tables A.2-A.5 present the detailed results for VRPTW, VRPMTW and VRPSTW benchmark instances. The first column shows the *Instance* name, the second column the previous *BKS* and the following columns show the average and best results achieved for each approach. The last rows present cumulated and averaged measures: Σ sums up the objective values over all instances, *Avg gap* shows the average gap to the previous BKS, *# BKS* counts the number of (current) BKS achieved with one approach, *Avg T* shows the average runtime, *CPU* the processor used and its *Passmark* score. **Bold** entries state the BKS, entries with an asterisk (*) are new BKS found by the HALNS. Further explanations are given below.

Additionally, Figures A.1 (VRPTW) and A.2 (VRPMTW) plot the average and best cumulated objective values against respective standardized runtimes, $\bar{T}_{\text{scaled}} [s]$, as if run on the AMD Ryzen 9 3900X using the passmark single thread ratings. The graphs of the HALNS are generated by varying the stopping condition, $gen^{\text{stop}} = [1, \dots, 10]$.

A.2.1 Results for Solomon and Desrosiers (1988) VRPTW Instances

Figure A.1 and Table A.2 show the detailed results for the VRPTW benchmark instances.

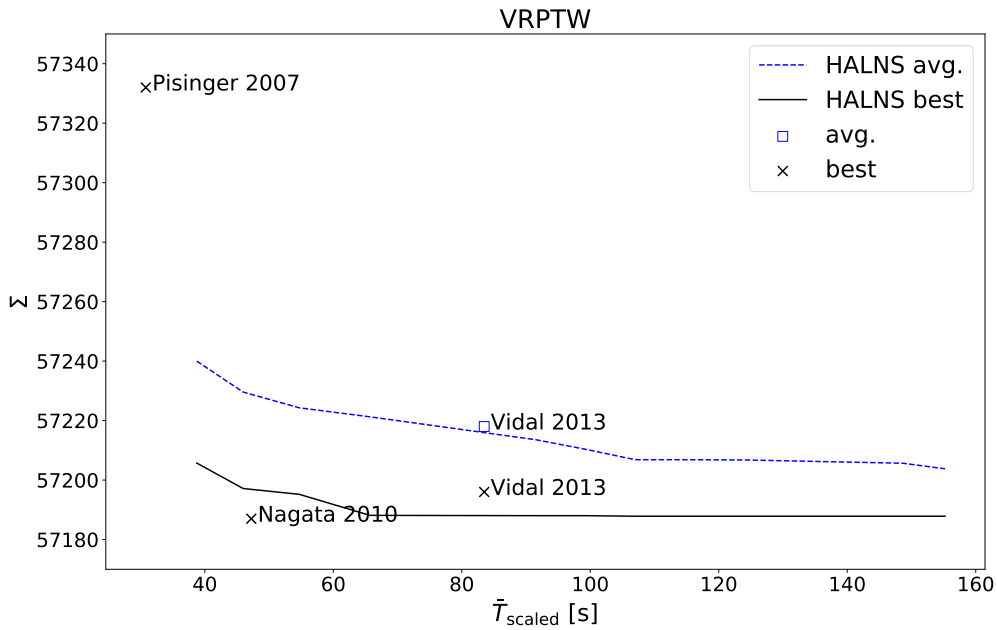


Figure A.1: VRPTW Benchmark

Appendix A Appendix for VRPAP

Table A.2: Detailed results for VRPTW instances

| Instance | BKS | HALNS | |
|--------------|---------|----------------|----------------|
| | | Best 10 | Avg 10 |
| C101.100.10 | 828.94 | 828.94 | 828.94 |
| C102.100.10 | 828.94 | 828.94 | 828.94 |
| C103.100.10 | 828.06 | 828.07 | 828.07 |
| C104.100.10 | 824.78 | 824.78 | 824.78 |
| C105.100.10 | 828.94 | 828.94 | 828.94 |
| C106.100.10 | 828.94 | 828.94 | 828.94 |
| C107.100.10 | 828.94 | 828.94 | 828.94 |
| C108.100.10 | 828.94 | 828.94 | 828.94 |
| C109.100.10 | 828.94 | 828.94 | 828.94 |
| C201.100.3 | 591.56 | 591.56 | 591.56 |
| C202.100.3 | 591.56 | 591.56 | 591.56 |
| C203.100.3 | 591.17 | 591.17 | 591.17 |
| C204.100.3 | 590.6 | 590.6 | 590.6 |
| C205.100.3 | 588.88 | 588.88 | 588.88 |
| C206.100.3 | 588.49 | 588.49 | 588.49 |
| C207.100.3 | 588.29 | 588.29 | 588.29 |
| C208.100.3 | 588.32 | 588.32 | 588.32 |
| R101.100.19 | 1650.8 | 1650.8 | 1650.8 |
| R102.100.17 | 1486.12 | 1486.12 | 1486.12 |
| R103.100.13 | 1292.68 | 1292.68 | 1292.68 |
| R104.100.9 | 1007.31 | 1007.31 | 1007.31 |
| R105.100.14 | 1377.11 | 1377.11 | 1377.11 |
| R106.100.12 | 1252.03 | 1252.03 | 1252.03 |
| R107.100.10 | 1104.66 | 1104.66 | 1104.74 |
| R108.100.9 | 960.88 | 960.88 | 961.44 |
| R109.100.11 | 1194.73 | 1194.73 | 1195.81 |
| R110.100.10 | 1118.84 | 1118.84 | 1118.9 |
| R111.100.10 | 1096.72 | 1096.73 | 1096.73 |
| R112.100.9 | 982.14 | 982.25 | 987.51 |
| R201.100.4 | 1252.37 | 1252.37 | 1252.37 |
| R202.100.3 | 1191.7 | 1191.7 | 1191.7 |
| R203.100.3 | 939.5 | 939.5 | 939.82 |
| R204.100.2 | 825.52 | 825.52 | 826.17 |
| R205.100.3 | 994.43 | 994.43 | 994.43 |
| R206.100.3 | 906.14 | 906.14 | 906.14 |
| R207.100.2 | 890.61 | 890.61 | 890.61 |
| R208.100.2 | 726.82 | 726.82 | 726.82 |
| R209.100.3 | 909.16 | 909.16 | 909.16 |
| R210.100.3 | 939.37 | 939.37 | 941.35 |
| R211.100.2 | 885.71 | 885.71 | 890.9 |
| RC101.100.14 | 1696.95 | 1696.95 | 1696.95 |
| RC102.100.12 | 1554.75 | 1554.75 | 1554.75 |
| RC103.100.11 | 1261.67 | 1261.67 | 1261.67 |
| RC104.100.10 | 1135.48 | 1135.48 | 1135.48 |
| RC105.100.13 | 1629.44 | 1629.44 | 1629.44 |
| RC106.100.11 | 1424.73 | 1424.73 | 1424.73 |
| RC107.100.11 | 1230.48 | 1230.48 | 1230.57 |
| RC108.100.10 | 1139.82 | 1139.82 | 1139.82 |
| RC201.100.4 | 1406.94 | 1406.94 | 1406.94 |
| RC202.100.3 | 1365.65 | 1365.65 | 1366.19 |
| RC203.100.3 | 1049.62 | 1050.45 | 1050.6 |
| RC204.100.3 | 798.46 | 798.46 | 798.46 |
| RC205.100.4 | 1297.65 | 1297.65 | 1297.65 |
| RC206.100.3 | 1146.32 | 1146.32 | 1146.32 |
| RC207.100.3 | 1061.14 | 1061.14 | 1061.14 |
| RC208.100.3 | 828.14 | 828.14 | 828.14 |
| Σ | 57187 | 57188 | 57204 |
| Avg gap | 0.00% | 0.00% | 0.03% |
| # BKS | 56 | | 52 |
| Avg T | | | 155s |
| CPU | | | Ryzen 9 3900X |
| Passmark | | | 2731 |

A.2.2 Results for Belhaiza et al. (2014) VRPMTW Instances

Figure A.2 and Table A.3 show the detailed results for the VRPMTW benchmark instances. Columns Belhaiza et al. (2014), Larsen and Pacino (2019), Schaap et al. (2019) and HALNS report the best and average results from 10 runs, while columns Belhaiza et al. (2017) and Hoogeboom et al. (2020) only show results from a single run. Columns K indicate the number of vehicles used in the corresponding best solution found.

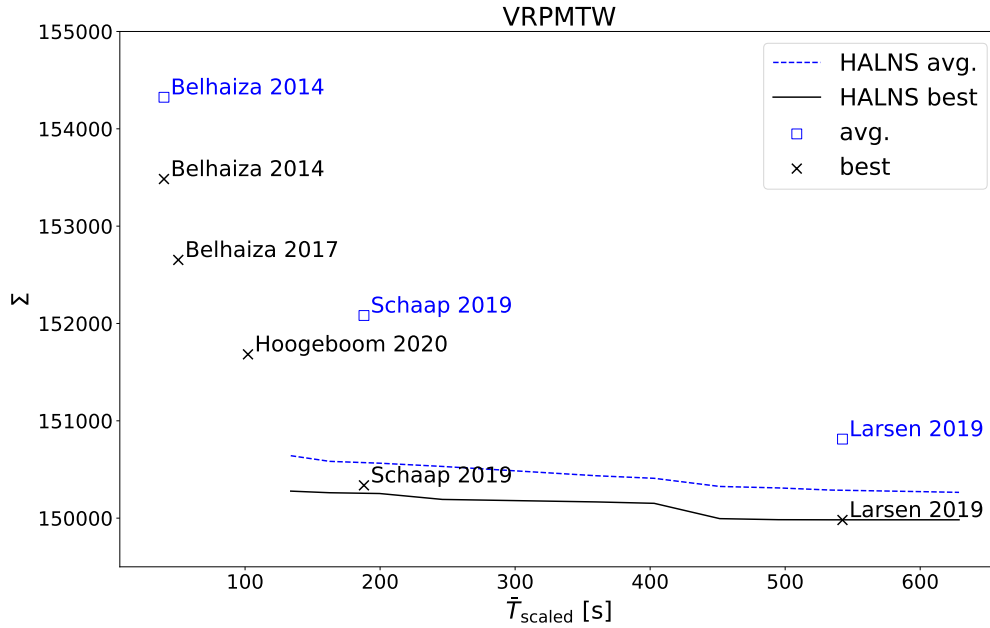


Figure A.2: VRPMTW Benchmark

Table A.3: Detailed Results for Belhaiza et al. (2014) VRPMTW Instances

| Instance | Belhaiza 14 | | Belhaiza 17 | | Larsen 19 | | Schaap 19 | | Hoogeboom 20 | | HALNS | | | |
|----------|-------------|---------------|---------------|---------------|---------------|---------------|---------------|--------|--------------|----------|---------------|-----|----------------|----------------|
| | K | Avg | K | Best/Avg | K | Avg | K | Best | K | Best/Avg | K | Avg | | |
| rm101 | 10 | 2977.2 | 3005 | 2970.2 | 10 | 2968.8 | 2969 | 2977.2 | 2991.9 | 10 | 2974.2 | 10 | 2971.3 | 2975.6 |
| rm102 | 9 | 2759.4 | 2759.4 | 2725.3 | 9 | 2705.9 | 2716.7 | 2730 | 2897.9 | 9 | 2726.7 | 9 | 2706.7 | 2713.2 |
| rm103 | 9 | 2692.5 | 2710.5 | 2681.5 | 9 | 2680.4 | 2681.6 | 2693.4 | 2702.8 | 9 | 2699 | 9 | 2680.5 | 2681.7 |
| rm104 | 9 | 2696.6 | 2719.2 | 2691.8 | 9 | 2690.7 | 2691.5 | 2697.6 | 2701.4 | 9 | 2690.7 | 9 | 2691.4 | 2692.6 |
| rm105 | 9 | 2688.8 | 2711 | 2687.2 | 9 | 2683.7 | 2683.7 | 2689.7 | 2699.9 | 9 | 2691.5 | 9 | 2683.7 | 2685.5 |
| rm106 | 9 | 2691.9 | 2691.9 | 2700.9 | 9 | 2700.9 | 2703.9 | 2707.1 | 2717.5 | 9 | 2700.9 | 9 | 2700.9 | 2700.9 |
| rm107 | 9 | 2690.8 | 2714.7 | 2685.1 | 9 | 2685.1 | 2685.1 | 2686 | 2700.4 | 9 | 2695.9 | 9 | 2685.1 | 2685.1 |
| rm108 | 9 | 2729.1 | 2729.1 | 2716 | 9 | 2716 | 2717.4 | 2727.6 | 2774.8 | 9 | 2725.5 | 9 | 2716 | 2716 |
| rm201 | 3 | 3711.4 | 3720.5 | 3686.4 | 2 | 2745.6 | 2756 | 2753.3 | 2780.6 | 2 | 2739.9 | 2 | 2738.4* | 2738.4* |
| rm202 | 2 | 2698.1 | 2717.3 | 2681 | 2 | 2681.3 | 2685 | 2691.3 | 2703.6 | 2 | 2679.9 | 2 | 2681.3 | 2681.3 |
| rm203 | 2 | 2686.1 | 2702 | 2673.6 | 2 | 2679 | 2679.7 | 2684.4 | 2693.6 | 2 | 2675.9 | 2 | 2677.9 | 2678.4 |
| rm204 | 2 | 2680.5 | 2691.2 | 2664.9 | 2 | 2672.8 | 2673.1 | 2678.4 | 2683.4 | 2 | 2671.7 | 2 | 2672.1 | 2674 |
| rm205 | 2 | 2671 | 2688.2 | 2651.3 | 2 | 2671 | 2671.1 | 2671 | 2678.8 | 2 | 2665.2 | 2 | 2671 | 2671.2 |
| rm206 | 2 | 2686.3 | 2704.9 | 2 | 2672.8 | 2679 | 2679 | 2688.3 | 2697.2 | 2 | 2665.6 | 2 | 2679 | 2679 |
| rm207 | 2 | 2678.2 | 2696.2 | 2 | 2657.3 | 2674.7 | 2678.2 | 2682.4 | 2693.8 | 2 | 2674.7 | 2 | 2674.4 | 2674.4 |
| rm208 | 2 | 2673.9 | 2690.7 | 2 | 2663.6 | 2673.9 | 2676.2 | 2678.6 | 2686 | 2 | 2662.2 | 2 | 2673.9 | 2674.3 |
| cm101 | 10 | 3089.2 | 3102.4 | 3101.2 | 10 | 3073 | 3126.8 | 3126.7 | 3172.6 | 10 | 3180.4 | 10 | 3144.6 | 3178.4 |
| cm102 | 12 | 3426.9 | 3426.9 | 3339.7 | 12 | 3392.5 | 3463 | 3467.8 | 3481.7 | 11 | 3433.6 | 11 | 3331.1* | 3436.8 |
| cm103 | 12 | 3532.7 | 3572.7 | 3520.5 | 11 | 3434.6 | 3477.1 | 3436.8 | 3471.9 | 11 | 3477.3 | 11 | 3435 | 3473.7 |
| cm104 | 14 | 4051.3 | 4058 | 4048 | 13 | 3859.6 | 3907.1 | 3870.8 | 3893.5 | 13 | 3862.6 | 13 | 3870.4 | 3885.5 |
| cm105 | 11 | 3060.6 | 3077.3 | 10 | 3010.2 | 3037.9 | 3051.7 | 3018.8 | 3061.7 | 10 | 3055.6 | 10 | 3044.1 | 3060.7 |
| cm106 | 10 | 2992.4 | 3020.2 | 10 | 2982.2 | 10 | 2980.2 | 2983.3 | 2999.8 | 10 | 2982.2 | 10 | 2980.3 | 2980.6 |
| cm107 | 11 | 3256.5 | 3292.3 | 11 | 3256.5 | 10 | 3077.9 | 2965.6 | 2968.6 | 10 | 2966.7 | 10 | 3077.9 | 3077.9 |
| cm108 | 10 | 2968.7 | 2973.1 | 10 | 2967.3 | 10 | 2965.4 | 2966.5 | 2968.6 | 10 | 2966.7 | 10 | 2965.1* | 2965.1* |
| cm201 | 5 | 4436.6 | 4452.5 | 5 | 4390.3 | 5 | 4361.9 | 4374.4 | 4419.6 | 5 | 4392.6 | 5 | 4377.7 | 4388 |
| cm202 | 6 | 4998.8 | 5024.9 | 6 | 4990.8 | 6 | 4982.4 | 4987.4 | 4996.5 | 6 | 4988.8 | 6 | 4982.4 | 4987.9 |
| cm203 | 5 | 4445.8 | 4484.6 | 5 | 4446.7 | 5 | 4440.3 | 4454.4 | 4463.8 | 5 | 4461.6 | 5 | 4440.4 | 4451.4 |
| cm204 | 5 | 4335.2 | 4372.4 | 5 | 4332.2 | 5 | 4317.5 | 4317.6 | 4335.7 | 5 | 4318.3 | 5 | 4317.5 | 4319.9 |
| cm205 | 4 | 3863.5 | 3883.2 | 4 | 3819.1 | 4 | 3816.8 | 3833.9 | 3838.4 | 4 | 3840.8 | 4 | 3816* | 3820.7 |
| cm206 | 4 | 3722 | 3743.2 | 4 | 3709.4 | 4 | 3699 | 3712 | 3713.1 | 4 | 3718.9 | 4 | 3699 | 3699 |
| cm207 | 4 | 3968.4 | 3977.8 | 4 | 3933.1 | 4 | 3917.4 | 3934.4 | 3941.7 | 4 | 3945.7 | 4 | 3917.4 | 3919 |
| cm208 | 4 | 3771.1 | 3793.2 | 4 | 3730.5 | 4 | 3714.6 | 3719.5 | 3738.3 | 4 | 3755.8 | 4 | 3713.9* | 3717.2 |
| rcm101 | 10 | 3062 | 3086.6 | 10 | 3063.5 | 10 | 3062 | 3062.2 | 3070.8 | 10 | 3063.5 | 10 | 3062.4 | 3063.3 |
| rcm102 | 10 | 3132.2 | 3142.3 | 10 | 3127.7 | 10 | 3110.8 | 3113.8 | 3157.2 | 10 | 3133.7 | 10 | 3110.8 | 3111.1 |
| rcm103 | 10 | 3152.9 | 3163.8 | 10 | 3131.8 | 10 | 3121.3 | 3121.9 | 3138.9 | 10 | 3152.1 | 10 | 3121.3 | 3121.8 |
| rcm104 | 10 | 3119.6 | 3134.6 | 10 | 3111.8 | 10 | 3111 | 3111.9 | 3124.1 | 10 | 3129.1 | 10 | 3111 | 3111 |
| rcm105 | 10 | 3187.9 | 3210.7 | 10 | 3185.9 | 10 | 3165.3 | 3169.5 | 3203.8 | 10 | 3187.8 | 10 | 3164.6* | 3166.2 |
| rcm106 | 10 | 3218.9 | 3218.9 | 10 | 3193.4 | 10 | 3158.2 | 3164.6 | 3173.1 | 10 | 3195.3 | 10 | 3158.2 | 3162.3 |
| rcm107 | 11 | 3488.9 | 3514 | 11 | 3487.7 | 11 | 3487.7 | 3490.2 | 3497.3 | 11 | 3488.9 | 11 | 3487.7 | 3487.7 |
| rcm108 | 11 | 3592.7 | 3592.7 | 11 | 3532.6 | 11 | 3531.1 | 3531.1 | 3544.7 | 11 | 3530.6 | 11 | 3529.5* | 3530.8 |
| rcm201 | 2 | 2804 | 2827.8 | 2 | 2778.7 | 2 | 2715.4 | 2748.1 | 2736.4 | 2 | 2706.8 | 2 | 2695.9* | 2695.9* |
| rcm202 | 2 | 2836.9 | 2846.8 | 2 | 2815.9 | 2 | 2716.8 | 2733.8 | 2749.1 | 2 | 2727.3 | 2 | 2716.3* | 2719.1 |
| rcm203 | 2 | 2721.9 | 2725.4 | 2 | 2722 | 2 | 2704.8 | 2714.2 | 2736.2 | 2 | 2825.5 | 2 | 2704.8 | 2704.9 |
| rcm204 | 2 | 2726.5 | 2743.1 | 2 | 2698.4 | 2 | 2692.6 | 2699.3 | 2703.9 | 2 | 2719.1 | 2 | 2690.5* | 2690.5* |
| rcm205 | 2 | 2754.5 | 2775.7 | 2 | 2754.5 | 2 | 2718.8 | 2725.1 | 2733.3 | 2 | 2724.2 | 2 | 2711.6* | 2712.3 |
| rcm206 | 2 | 2812.7 | 2830.6 | 2 | 2769.6 | 2 | 2721.9 | 2743.1 | 2764.1 | 2 | 2781.3 | 2 | 2721.9 | 2722.9 |
| rcm207 | 3 | 3749.8 | 3786.8 | 3 | 3749.8 | 2 | 2861 | 3216.9 | 3585.4 | 3 | 3744.9 | 2 | 2857.6* | 2857.6* |

Continued on next page

Table A.3 – Continued from previous page

| | 2 | 2791.4 | 2817.2 | 2 | 2742.7 | 2 | 2722.7 | 2732.5 | 2 | 2722.7 | 2755.1 | 2 | 2722.7 | 2735.1 | 2 | 2722.7 | 2724.2 |
|----------|------------|--------|--------|------------|--------|----------|--------|--------|-----------|--------|--------|----------|--------|----------|-------|---------------|--------|
| rcm208 | 2 | 2791.4 | 2817.2 | 2 | 2742.7 | 2 | 2722.7 | 2732.5 | 2 | 2722.7 | 2755.1 | 2 | 2722.7 | 2735.1 | 2 | 2722.7 | 2724.2 |
| Σ | 316 | 153484 | 154324 | 314 | 152653 | 310 | 149981 | 150812 | 310 | 150337 | 152082 | 310 | 150337 | 151683 | 309 | 149983 | 150265 |
| Avg gap | 2.60% | 3.16% | | 2.04% | | 0.13% | 0.70% | | 0.37% | 1.60% | | 0.37% | 1.33% | | 0.13% | 0.31% | |
| # BKS | 2 | | | 8 | | 27 | | | 5 | | | 5 | | 5 | | 28 (12 new) | |
| Avg T | 64s | | | 81s | | 600s | | | 185s | | | 113s | | 113s | | 629s | |
| CPU | i5 3.3 GHz | | | i5 3.3 GHz | | i7-4790K | | | i7 3.7GHz | | | i7 4 GHz | | i7 4 GHz | | Ryzen 9 3900X | |
| Passm. | 1704 | | | 1704 | | 2469 | | | 2776 | | | 2469 | | 2469 | | 2731 | |

A.2.3 Results for VRPSTW Instances

We compare our results for the VRPSTW against the approaches of Vidal et al. (2014b), Mouthuy et al. (2015) and Kritzinger et al. (2017). Tables A.4 and A.5 present the detailed results for types 1 and 2 with $\alpha = 1$ as described above.

Table A.4: Detailed results for VRPSTW instances type 1 (only lateness), $\alpha = 1$

| Instance | BKS | Kritzinger | | Vidal | | HALNS | |
|--------------|----------|----------------|---------------|----------------|----------------|----------------|----------------|
| | | Best 10 | Avg 10 | Best 10 | Avg 10 | Best 10 | Avg 10 |
| C101.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C102.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C103.100.10 | 828.06 | 828.07 | 828.07 | 828.06 | 828.06 | 828.07 | 828.07 |
| C104.100.10 | 824.78 | 824.78 | 824.78 | 824.78 | 824.78 | 824.78 | 824.78 |
| C105.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C106.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C107.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C108.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C109.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C201.100.3 | 591.56 | 591.56 | 591.56 | 591.56 | 591.56 | 591.56 | 591.56 |
| C202.100.3 | 591.56 | 591.56 | 591.56 | 591.56 | 591.56 | 591.56 | 591.56 |
| C203.100.3 | 591.17 | 591.17 | 591.17 | 591.17 | 591.17 | 591.17 | 591.17 |
| C204.100.3 | 590.6 | 590.6 | 590.6 | 590.6 | 590.6 | 590.6 | 590.6 |
| C205.100.3 | 588.88 | 588.88 | 588.88 | 588.88 | 588.88 | 588.88 | 588.88 |
| C206.100.3 | 588.49 | 588.49 | 588.49 | 588.49 | 588.49 | 588.49 | 588.49 |
| C207.100.3 | 588.29 | 588.29 | 588.29 | 588.29 | 588.29 | 588.29 | 588.29 |
| C208.100.3 | 588.32 | 588.32 | 588.32 | 588.32 | 588.32 | 588.32 | 588.32 |
| R101.100.19 | 1562.58 | 1562.58 | 1562.98 | 1562.58 | 1562.89 | 1562.58 | 1562.58 |
| R102.100.17 | 1379.11 | 1379.11 | 1379.62 | 1379.11 | 1379.21 | 1379.11 | 1379.11 |
| R103.100.13 | 1159.28 | 1159.54 | 1160.64 | 1159.28 | 1159.51 | 1159.28 | 1159.41 |
| R104.100.9 | 999.77 | 1003.73 | 1009.02 | 999.77 | 999.77 | 999.77 | 1000.04 |
| R105.100.14 | 1347.75 | 1347.75 | 1348.89 | 1347.75 | 1347.75 | 1347.75 | 1347.75 |
| R106.100.12 | 1236.58 | 1236.58 | 1237.29 | 1236.58 | 1236.58 | 1236.58 | 1236.6 |
| R107.100.10 | 1083.62 | 1084.96 | 1089.84 | 1083.62 | 1083.62 | 1083.62 | 1083.66 |
| R108.100.9 | 946.6 | 949.94 | 951.24 | 946.6 | 947.04 | 946.6 | 947.22 |
| R109.100.11 | 1173.21 | 1173.21 | 1176.4 | 1173.21 | 1173.21 | 1173.21 | 1173.21 |
| R110.100.10 | 1106.66 | 1106.66 | 1114.66 | 1107.26 | 1111.57 | 1109.58 | 1110.22 |
| R111.100.10 | 1074.84 | 1080.25 | 1086.36 | 1074.84 | 1076.41 | 1074.84 | 1077.03 |
| R112.100.9 | 971.31 | 972.11 | 981.82 | 971.31 | 975.78 | 971.31 | 971.94 |
| R201.100.4 | 1237.11 | 1237.11 | 1237.17 | 1237.11 | 1237.11 | 1237.11 | 1237.11 |
| R202.100.3 | 1165.32 | 1165.32 | 1169.23 | 1165.32 | 1165.32 | 1165.32 | 1165.32 |
| R203.100.3 | 933.52 | 937.35 | 942.96 | 933.52 | 934.01 | 934.1 | 934.1 |
| R204.100.2 | 824.02 | 832.38 | 840.79 | 824.02 | 824.73 | 824.02 | 825.4 |
| R205.100.3 | 994.43 | 994.43 | 1006.79 | 994.43 | 994.43 | 994.43 | 994.43 |
| R206.100.3 | 906.14 | 912.81 | 920.13 | 906.14 | 906.14 | 906.14 | 906.14 |
| R207.100.2 | 887.28 | 908.7 | 1044.87 | 887.28 | 888.44 | 887.28 | 887.28 |
| R208.100.2 | 726.82 | 728.92 | 735.26 | 726.82 | 727.08 | 726.82 | 726.82 |
| R209.100.3 | 909.16 | 909.3 | 917.21 | 909.16 | 909.16 | 909.16 | 909.31 |
| R210.100.3 | 938.34 | 948.8 | 958.58 | 938.34 | 941.95 | 938.34 | 938.34 |
| R211.100.2 | 885.71 | 901.18 | 923.85 | 885.71 | 892.5 | 890.79 | 890.79 |
| RC101.100.14 | 1590.22 | 1590.22 | 1591.59 | 1590.22 | 1590.22 | 1590.22 | 1591.23 |
| RC102.100.12 | 1428.21 | 1428.21 | 1429.9 | 1428.21 | 1428.21 | 1428.21 | 1428.21 |
| RC103.100.11 | 1239.54 | 1239.54 | 1242.33 | 1239.54 | 1239.73 | 1239.54 | 1240.91 |
| RC104.100.10 | 1126.31 | 1126.31 | 1128.74 | 1126.31 | 1126.31 | 1126.31 | 1126.31 |
| RC105.100.13 | 1450.84 | 1450.84 | 1451.38 | 1450.84 | 1450.84 | 1450.84 | 1450.84 |
| RC106.100.11 | 1349.3 | 1349.3 | 1350.17 | 1349.3 | 1349.72 | 1350.57 | 1353.63 |
| RC107.100.11 | 1208.81 | 1208.81 | 1208.96 | 1208.81 | 1208.98 | 1208.81 | 1208.81 |
| RC108.100.10 | 1118 | 1118 | 1119.61 | 1118 | 1118.31 | 1119.59 | 1119.59 |
| RC201.100.4 | 1380.33 | 1380.33 | 1380.47 | 1380.33 | 1380.33 | 1380.33 | 1380.33 |
| RC202.100.3 | 1317.28 | 1317.28 | 1322.17 | 1317.28 | 1317.28 | 1317.28 | 1317.37 |
| RC203.100.3 | 1040.77 | 1046.05 | 1057.1 | 1040.77 | 1045 | 1040.77 | 1040.77 |
| RC204.100.3 | 797.04 | 797.41 | 809.09 | 797.04 | 797.04 | 797.04 | 797.04 |
| RC205.100.4 | 1297.65 | 1299.61 | 1305.97 | 1297.65 | 1298 | 1297.65 | 1299.08 |
| RC206.100.3 | 1135.26 | 1135.26 | 1135.9 | 1135.26 | 1135.26 | 1135.26 | 1135.26 |
| RC207.100.3 | 1056.88 | 1061.14 | 1073.58 | 1056.88 | 1058.16 | 1056.88 | 1056.88 |
| RC208.100.3 | 827.67 | 829 | 834.82 | 827.67 | 827.9 | 827.67 | 827.67 |
| Σ | 55987.56 | 56084.33 | 56411.68 | 55988.16 | 56019.79 | 55999.01 | 56012.04 |
| Avg gap | 0.00% | 0.19% | 0.81% | 0.00% | 0.06% | 0.02% | 0.04% |
| # BKS | 56/56 | 36 | | 55 | | 50 | |
| Avg T | | 600s | | 349s | | 156s | |
| CPU | | Xeon E7-8837 | | Opteron 2.2G | | Ryzen 9 3900X | |
| Passmark | | 1124 | | 445 | | 2731 | |

Appendix A Appendix for VRPAP

Table A.5: Detailed results for VRPSTW instances type 2 (earliness and lateness), $\alpha = 1$

| Instance | BKS | Vidal | | HALNS | |
|--------------|---------|----------------|----------------|-----------------|-----------------|
| | | Best 10 | Avg 10 | Best 10 | Avg 10 |
| C101.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C102.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C103.100.10 | 828.06 | 828.06 | 828.06 | 828.07 | 828.07 |
| C104.100.10 | 824.78 | 824.78 | 824.78 | 824.78 | 824.78 |
| C105.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C106.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C107.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C108.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C109.100.10 | 828.94 | 828.94 | 828.94 | 828.94 | 828.94 |
| C201.100.3 | 591.56 | 591.56 | 591.56 | 591.56 | 591.56 |
| C202.100.3 | 591.56 | 591.56 | 591.56 | 591.56 | 591.56 |
| C203.100.3 | 591.17 | 591.17 | 591.17 | 591.17 | 591.17 |
| C204.100.3 | 590.6 | 590.6 | 590.6 | 590.6 | 590.6 |
| C205.100.3 | 588.88 | 588.88 | 588.88 | 588.88 | 588.88 |
| C206.100.3 | 588.49 | 588.49 | 588.49 | 588.49 | 588.49 |
| C207.100.3 | 588.29 | 588.29 | 588.29 | 588.29 | 588.29 |
| C208.100.3 | 588.32 | 588.32 | 588.32 | 588.32 | 588.32 |
| R101.100.19 | 1546.91 | 1546.91 | 1546.91 | 1557.82 | 1561.92 |
| R102.100.17 | 1377.38 | 1377.38 | 1377.38 | 1378.44 | 1379.8 |
| R103.100.13 | 1158.31 | 1158.31 | 1158.83 | 1158.48 | 1159.71 |
| R104.100.9 | 1000.33 | 1000.33 | 1004.57 | 1002.68 | 1003.7 |
| R105.100.14 | 1342.57 | 1342.57 | 1342.57 | 1347.61 | 1347.72 |
| R106.100.12 | 1223.09 | 1223.09 | 1223.09 | 1226.13 | 1226.8 |
| R107.100.10 | 1079.12 | 1079.12 | 1080.9 | 1081.51 | 1083.76 |
| R108.100.9 | 945.64 | 945.64 | 948.23 | 944.23* | 947.33 |
| R109.100.11 | 1164.68 | 1164.68 | 1164.68 | 1167.88 | 1173.72 |
| R110.100.10 | 1104.59 | 1104.59 | 1108.3 | 1107.34 | 1111.01 |
| R111.100.10 | 1065.76 | 1065.76 | 1065.76 | 1066.47 | 1070.24 |
| R112.100.9 | 969.91 | 969.91 | 991.5 | 969.98 | 970.25 |
| R201.100.4 | 1235.14 | 1235.14 | 1235.14 | 1237.11 | 1237.11 |
| R202.100.3 | 1159.76 | 1159.76 | 1159.76 | 1165.32 | 1165.32 |
| R203.100.3 | 934.1 | 934.1 | 937.04 | 934.09* | 934.09* |
| R204.100.2 | 820.9 | 820.9 | 837.21 | 821.54 | 821.54 |
| R205.100.3 | 994.43 | 994.43 | 996.24 | 994.43 | 994.43 |
| R206.100.3 | 906.54 | 906.54 | 910.99 | 906.14* | 906.14* |
| R207.100.2 | 906.81 | 906.81 | 937.79 | 887.28* | 891.01 |
| R208.100.2 | 730.52 | 730.52 | 735.31 | 726.82* | 727.18 |
| R209.100.3 | 909.16 | 909.16 | 911.61 | 909.16 | 909.16 |
| R210.100.3 | 938.77 | 938.77 | 948.91 | 938.34* | 938.34* |
| R211.100.2 | 912.39 | 912.39 | 921.81 | 885.71* | 890.26 |
| RC101.100.14 | 1584.2 | 1584.2 | 1584.2 | 1588.58 | 1591.72 |
| RC102.100.12 | 1409.36 | 1409.36 | 1409.36 | 1416.67 | 1417.69 |
| RC103.100.11 | 1231.67 | 1231.67 | 1231.67 | 1239.54 | 1241.41 |
| RC104.100.10 | 1121.84 | 1121.84 | 1123.25 | 1126.31 | 1127.07 |
| RC105.100.13 | 1433.37 | 1433.37 | 1433.37 | 1436.99 | 1437.04 |
| RC106.100.11 | 1334.89 | 1334.89 | 1334.89 | 1338.42 | 1338.49 |
| RC107.100.11 | 1203.06 | 1203.06 | 1203.06 | 1207.04 | 1208.39 |
| RC108.100.10 | 1115.44 | 1115.44 | 1115.44 | 1119.59 | 1119.84 |
| RC201.100.4 | 1380.33 | 1380.33 | 1380.33 | 1380.33 | 1380.33 |
| RC202.100.3 | 1312.05 | 1312.05 | 1312.05 | 1312.05 | 1312.06 |
| RC203.100.3 | 1044.74 | 1044.74 | 1047.43 | 1040.65* | 1040.65* |
| RC204.100.3 | 796.68 | 796.68 | 796.91 | 797.04 | 797.05 |
| RC205.100.4 | 1297.86 | 1297.86 | 1300.98 | 1297.97 | 1298.78 |
| RC206.100.3 | 1135.26 | 1135.26 | 1135.44 | 1135.26 | 1135.26 |
| RC207.100.3 | 1056.88 | 1056.88 | 1061.92 | 1056.88 | 1056.88 |
| RC208.100.3 | 827.67 | 827.67 | 832.3 | 827.67 | 827.67 |
| Σ | 55886.4 | 55886.4 | 56021.42 | 55909.8 | 55955.17 |
| Avg gap | 0.00% | 0.00% | 0.26% | 0.00% | 0.07% |
| # BKS | 48/56 | | 48 | | 31 (8 new) |
| Avg T | | | 1797s | | 159s |
| CPU | | | Opteron 2.2G | | Ryzen 9 3900X |
| Passmark | | | 445 | | 2731 |

A.2.4 Results for VRPAP Instances

Table A.6 compares the HALNS algorithm with the lower and upper bound described in Section 2.4. Column C_{VRP}^{trans} shows the lowest possible transportation costs, achieved by solving a VRP, Column C_{VRPTW}^{failed} the lowest possible failed-delivery costs, and Column C_{VRPAP}^{lower} the lower bound, derived by the summation of both terms before. Column *Best 10* and *Avg 10* show the best and average result of ten runs. Column C_{VRPAP}^{upper} signifies the upper bound. Columns Δ^{lb} and Δ^{ub} represent the percentage gaps to the lower and upper bound, respectively. Column *Avg T [s]* shows the average runtime of ten runs in seconds.

Table A.6: Results for VRPAP instances

| Instance | C_{VRP}^{trans} | C_{VRPTW}^{failed} | C_{VRPAP}^{lower} | Best 10 | Avg 10 | C_{VRPAP}^{upper} | Δ^{lb} | Δ^{ub} | Avg T [s] |
|--------------|-------------------|----------------------|---------------------|---------|---------|---------------------|---------------|---------------|-----------|
| R101 | | | | | | | | | |
| A | 865.95 | 86.6 | 952.55 | 1213.82 | 1217 | 1241.08 | 27% | -2% | 811 |
| V | 865.95 | 86.6 | 952.55 | 1279.87 | 1286.43 | 1355.9 | 34% | -6% | 779 |
| W | 865.95 | 173.19 | 1039.14 | 1253.02 | 1260.39 | 1326.98 | 21% | -6% | 1020 |
| M | 865.95 | 86.6 | 952.55 | 1191.85 | 1198.11 | 1269.92 | 25% | -6% | 819 |
| AV | 865.95 | 86.6 | 952.55 | 1216.4 | 1223.22 | 1303.43 | 28% | -7% | 1044 |
| WM | 865.95 | 129.89 | 995.84 | 1163.34 | 1168.37 | 1304.21 | 17% | -11% | 1001 |
| AVWM | 865.95 | 108.24 | 974.19 | 1180.49 | 1188.29 | 1298.67 | 21% | -9% | 1068 |
| R201 | | | | | | | | | |
| A | 651.3 | 65.13 | 716.43 | 915.79 | 920.04 | 936.57 | 28% | -2% | 1074 |
| V | 651.3 | 65.13 | 716.43 | 919.24 | 920.19 | 1012.77 | 28% | -9% | 1047 |
| W | 651.3 | 130.26 | 781.56 | 907.79 | 910.85 | 987.63 | 16% | -8% | 1043 |
| M | 651.3 | 65.13 | 716.43 | 919.76 | 921.02 | 962.04 | 28% | -4% | 979 |
| AV | 651.3 | 65.13 | 716.43 | 887.67 | 888.56 | 967.83 | 24% | -8% | 968 |
| WM | 651.3 | 97.7 | 749 | 882.62 | 882.91 | 976.62 | 18% | -10% | 969 |
| AVWM | 651.3 | 81.41 | 732.71 | 888.17 | 889.87 | 968.94 | 21% | -8% | 1023 |
| C101 | | | | | | | | | |
| A | 824.78 | 82.48 | 907.26 | 1141.24 | 1141.41 | 1187.68 | 26% | -4% | 699 |
| V | 824.78 | 82.48 | 907.26 | 1167.89 | 1169.7 | 1287.65 | 29% | -9% | 681 |
| W | 824.78 | 164.96 | 989.74 | 1149.36 | 1152.89 | 1216.96 | 16% | -6% | 929 |
| M | 824.78 | 82.48 | 907.26 | 1125.82 | 1125.82 | 1255.81 | 24% | -10% | 769 |
| AV | 824.78 | 82.48 | 907.26 | 1106.09 | 1107.16 | 1225.29 | 22% | -10% | 958 |
| WM | 824.78 | 123.72 | 948.5 | 1054.43 | 1057.28 | 1257.05 | 11% | -16% | 1009 |
| AVWM | 824.78 | 103.1 | 927.88 | 1062.39 | 1063.63 | 1239.23 | 14% | -14% | 871 |
| C201 | | | | | | | | | |
| A | 584.28 | 58.43 | 642.71 | 857.86 | 867.97 | 862.51 | 33% | -1% | 1129 |
| V | 584.28 | 58.43 | 642.71 | 860.06 | 861.97 | 889.98 | 34% | -3% | 764 |
| W | 584.28 | 116.86 | 701.14 | 856.55 | 858.52 | 885.89 | 22% | -3% | 814 |
| M | 584.28 | 58.43 | 642.71 | 857.72 | 857.91 | 866.95 | 33% | -1% | 838 |
| AV | 584.28 | 58.43 | 642.71 | 864.48 | 870.26 | 885.89 | 35% | -2% | 860 |
| WM | 584.28 | 87.64 | 671.92 | 832.31 | 835.94 | 881.27 | 24% | -6% | 789 |
| AVWM | 584.28 | 73.04 | 657.32 | 851.95 | 855.96 | 892.6 | 30% | -5% | 996 |
| RC101 | | | | | | | | | |
| A | 995.59 | 99.56 | 1095.15 | 1361.97 | 1363.45 | 1396.22 | 24% | -2% | 904 |
| V | 995.59 | 99.56 | 1095.15 | 1519.48 | 1521.66 | 1588.96 | 39% | -4% | 1148 |
| W | 995.59 | 199.12 | 1194.71 | 1438.19 | 1444.22 | 1521.06 | 20% | -5% | 990 |
| M | 995.59 | 99.56 | 1095.15 | 1383.98 | 1391.03 | 1468.79 | 26% | -6% | 1134 |
| AV | 995.59 | 99.56 | 1095.15 | 1393.96 | 1397.17 | 1475.27 | 27% | -6% | 1273 |
| WM | 995.59 | 149.34 | 1144.93 | 1332.51 | 1333.9 | 1442.61 | 16% | -8% | 952 |
| AVWM | 995.59 | 124.45 | 1120.04 | 1393.37 | 1399.14 | 1482.93 | 24% | -6% | 766 |
| RC201 | | | | | | | | | |
| A | 655.35 | 65.54 | 720.89 | 933.33 | 938.44 | 944.36 | 29% | -1% | 1113 |
| V | 655.35 | 65.54 | 720.89 | 947.69 | 951.66 | 1022.61 | 31% | -7% | 1031 |
| W | 655.35 | 131.07 | 786.42 | 937.76 | 942.06 | 1006.55 | 19% | -7% | 1043 |
| M | 655.35 | 65.54 | 720.89 | 931.21 | 932.37 | 959.17 | 29% | -3% | 805 |
| AV | 655.35 | 65.54 | 720.89 | 917.57 | 919.01 | 949.6 | 27% | -3% | 787 |
| WM | 655.35 | 98.3 | 753.65 | 907.02 | 908.96 | 979.94 | 20% | -7% | 964 |
| AVWM | 655.35 | 81.92 | 737.27 | 918.96 | 919.17 | 960.09 | 25% | -4% | 931 |
| Avg | 762.88 | 95.36 | 858.23 | 1069.64 | 1072.95 | 1141.56 | 25% | -6% | 943 |

A.3 Results for Components

Figure A.3 plots the average objective values of ten HALNS runs as a function of runtime when deactivating one component at a time (or activating local improvement, respectively) as described in Section 2.6.2.3. As before, the graphs are generated by varying the stopping condition, $gen^{stop} = [1, \dots, 10]$.

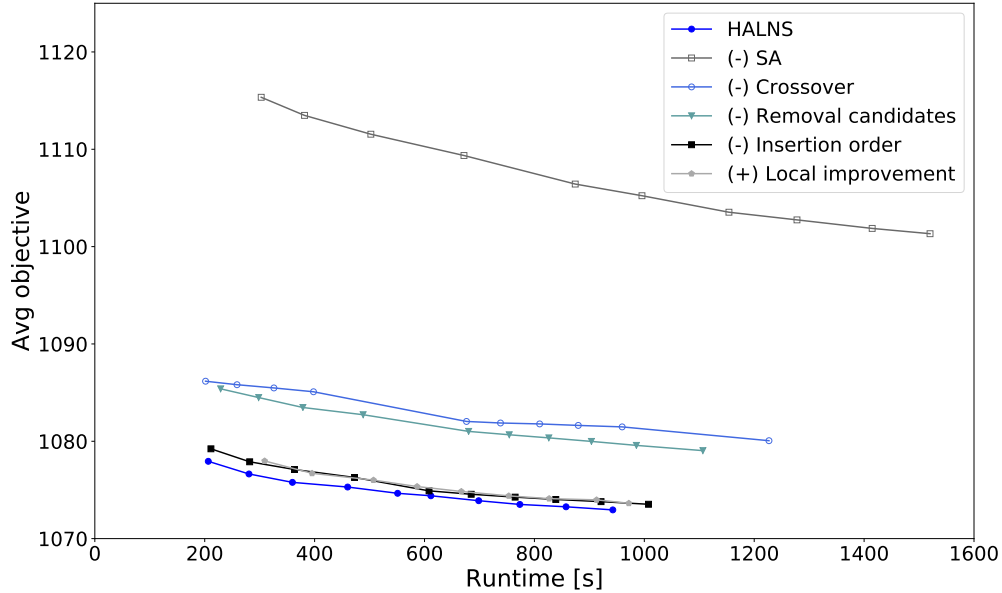


Figure A.3: HALNS performance with and without specific components for VRPAP instances

A.4 UKTUS Availability Profiles

Figure A.4 shows the average APs depending on age and employment status. The x axis shows the time from 8-18 o'clock, while the y axis represents the average availability in percent. It can be observed that the average availability for unemployed people is generally higher compared to employed people and that average availability increases with age in most cases. Furthermore the profiles up to age 60 represent a V-profile with highest availability in the morning and evening hours, whereas elderly people seem to return home during midday and leave again in the afternoon, resulting in a W-profile.

Appendix A Appendix for VRPAP

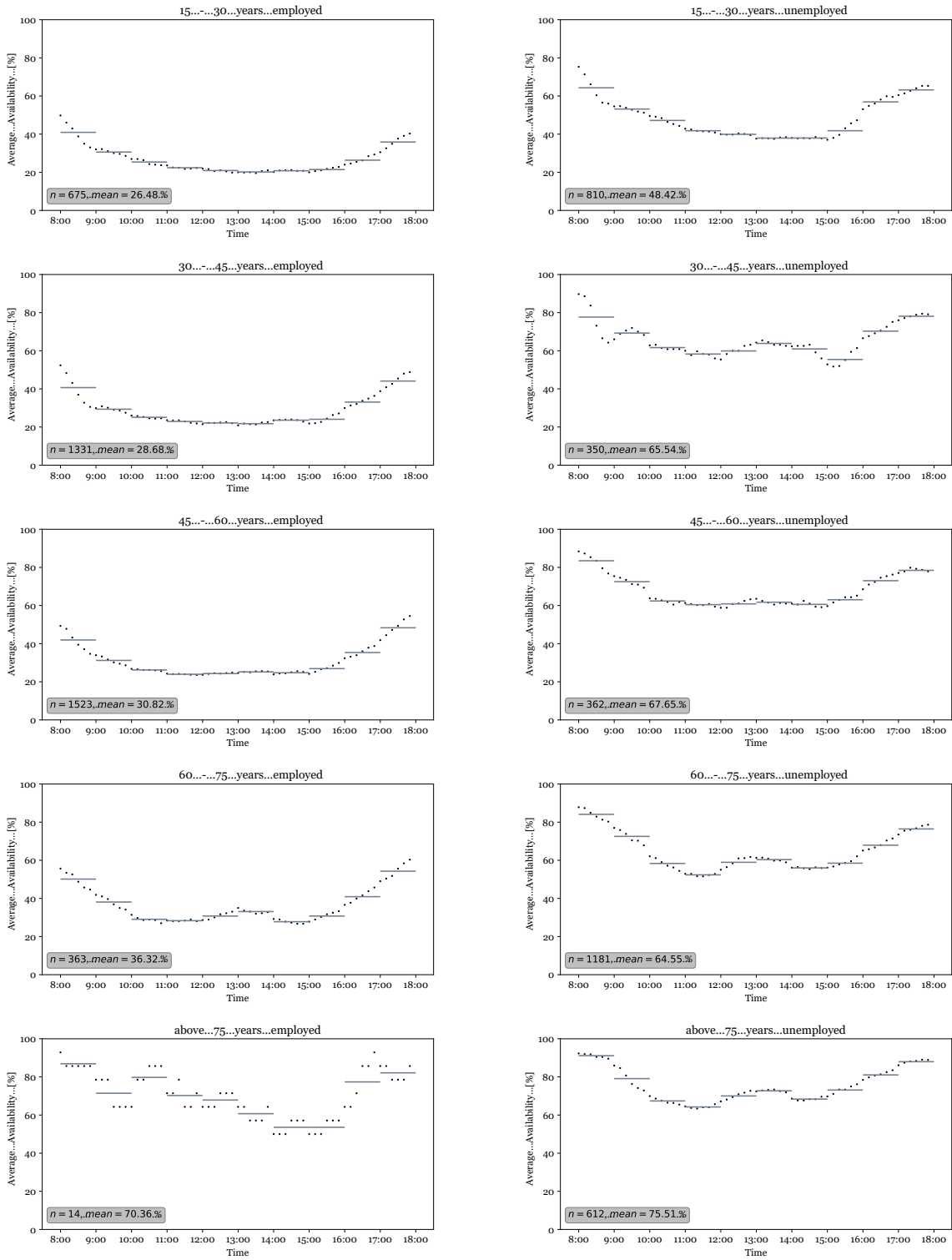


Figure A.4: UKTUS Availability Profiles depending on Age and Work Status

Appendix B

Appendix for Hybrid Adaptive Large Neighborhood Search for Vehicle Routing Problems with Depot Location Decisions

B.1 Parameters

Table B.1: Parameters for HALNS

| Parameter | Meaning | Chosen Value |
|--------------------|--|--|
| n^P | Size of the initial population | 12 |
| gen^{\max} | Maximum number of generations | 10 in Section 3.5.2 100 in Section 3.5.3 50 in Section 3.5.4 |
| it^{stop} | Number of ALNS iterations without improvement | 10,000 |
| α | Cool rate in SA | 0.9997 |
| χ_0 | Acceptance probability in SA | 0.10 |
| n_0 | Number of iterations for determining the initial SA temperature | 400 |
| ω | Weight for penalties drawn at the beginning of every ALNS run | $unif(0.0001, 0.004) \cdot f(\hat{s})$ |
| p^{binom} | Probability for binomial distribution drawn at the beginning of every ALNS run | $unif(0.10, 0.35)$ |
| σ_1 | Score for new best solution | 35 |
| σ_2 | Score for new best current solution | 2 |
| σ_3 | Score for worse solution, but accepted via SA | 1 |
| γ | Reaction factor for adapting scores for operators and p^{binom} | 0.3 |

B.2 Scaling of Runtimes

Table B.2 tests whether runtimes can reasonably be standardized by using the passmark single thread rating. The table shows the runtimes of the HALNS ($n^P = 12$, $gen^{\max} = 50$) on both CPUs (AMD Ryzen 3900X and AMD Ryzen 2700X) for all three problem classes. The last column shows the actual runtime reduction [%]. The passmark single thread rating seems to be a reasonably good approximation for standardizing runtimes, as the runtime of running all instances is reduced to 91.4% when running on the 3900X vs. 89.3% when estimated via passmark rating.

Table B.3 lists the used CPUs with their respective passmark single thread rating (<https://www.cpubenchmark.net/>). Mühlbauer and Fontaine (2021) use 4 threads; we therefore multiply the runtimes by four (1). Not all authors clearly state the CPU used. In these cases we made an

Table B.2: Analysis of Runtime Scaling Factor

| Problem Class | Runtime [s] | | Runtime [%] |
|---------------|-----------------|-----------------|-------------|
| | AMD Ryzen 3900X | AMD Ryzen 2700X | |
| 2E-VRP | 214 | 226 | 94.7 |
| LRP | 245 | 290 | 84.5 |
| MDVRP | 510 | 556 | 91.7 |
| All | 252 | 276 | 91.4 |

assumption based on the information given (2). Vidal et al. (2012) use an AMD Opteron 250 with 2.4GHz, but report their runtimes as if run on an Intel Pentium 4 with 3 GHz (3). No passmark rating could be found for a small number of CPUs used (4).

Table B.3: CPUs used

| Author | CPU | Clock Speed | Rating | Notes |
|---------------------------------|------------------------|-------------|--------|-------|
| Voigt et al. | AMD Ryzen 9 3900X | 3.8 GHz | 2731 | |
| Voigt et al. | AMD Ryzen 7 2700X | 3.7 GHz | 2439 | |
| Wang et al. (2017) | Intel Xeon E5-2670 | 2.6 GHz | 1460 | |
| Amarouche et al. (2018) | Intel Xeon E5-2670v2 | 2.5 GHz | 1592 | |
| Mühlbauer and Fontaine (2021) | Intel i5-6200 | 2.4 GHz | 1600 | 1 |
| Breunig et al. (2016) | Intel Xeon E5-2670v2 | 2.5 GHz | 1592 | |
| Hemmelmayr et al. (2012) | AMD Opteron 275 | 2.2 GHz | 445 | |
| Enthoven et al. (2020) | Intel Xeon E5-2680v3 | 2.5 GHz | 1774 | |
| Yu et al. (2021) | Intel i7-8700 | 3.2 GHz | 2666 | |
| Schneider and Löffler (2019) | Intel Xeon E5-2670 | 2.6 GHz | 1460 | |
| Arnold and Sörensen (2021) | AMD Ryzen 3 1300X | 3.5 GHz | 2084 | |
| Contardo et al. (2013) | Intel Xeon E5462 | 3.0 GHz | 1215 | |
| Hemmelmayr et al. (2012) | AMD Opteron 275 | 2.2 GHz | 445 | |
| Lopes et al. (2016) | Intel i7-4790 | 3.6 GHz | 2226 | |
| Escobar et al. (2014a) | Intel Core2 Duo T6400 | 2.0 GHz | 799 | 2 |
| Ting and Chen (2013) | AMD Athlon XP 2500+ | 1.83 GHz | 353 | |
| Escobar et al. (2013) | Intel Core2 Duo T6400 | 2.0 GHz | 799 | 2 |
| Yu et al. (2010) | Intel Core2 Quad Q8400 | 2.6 GHz | 1152 | 2 |
| Duhamel et al. (2010) | Intel Core2 Quad Q9550 | 2.83 GHz | 1228 | 2 |
| Prins et al. (2007) | Intel Pentium 4 | 2.4 GHz | 360 | |
| Prins et al. (2006a) | Intel Pentium 4 | 2.4 GHz | 360 | |
| Prins et al. (2006b) | Intel Pentium 4 | 2.4 GHz | 360 | |
| Vidal et al. (2012) | Intel Pentium 4 | 3.0 GHz | 561 | 3 |
| Arnold and Sörensen (2019) | AMD Ryzen 3 1300X | 3.5 GHz | 2084 | |
| Luo and Chen (2014) | Intel Pentium 4 | 2.8 GHz | 506 | 2 |
| Subramanian et al. (2013) | Intel Core i7 | 2.93 GHz | 1403 | |
| Pisinger and Ropke (2007) | Intel Pentium 4 | 3.0 GHz | 561 | |
| Sadati et al. (2021) | Intel Core i7-8700 | 3.2 GHz | 2666 | |
| Cordeau and Maischberger (2012) | Intel Xeon X7350 | 2.93 GHz | 1181 | |
| Escobar et al. (2014b) | Intel Core2 Duo T6400 | 2.0 GHz | 799 | 2 |
| Renaud et al. (1996) | Sun SparcStation 10 | NA | 100 | 4 |
| Cordeau et al. (1997) | Sun SparcStation 10 | NA | 100 | 4 |
| Chao et al. (1993) | Sun 4/370 | NA | 100 | 4 |

B.3 Detailed Results on Benchmark Instances

Tables B.4 - B.6 present the detailed results for 2E-VRP, LRP and MDVRP benchmark instances. The first column shows the *Instance* name, the second column the (previous) *BKS*, column *Best 5/10* the best result obtained after five or ten runs of the HALNS, and *Avg 5/10* the average results

Appendix B Appendix for HALNS for VRPs with Depot Location Decisions

across five or ten runs. *Runtime [s]* shows the average runtime in seconds of the HALNS run on the AMD Ryzen 9 3900X. The second-last row presents the *Avg gap* to the previous BKS and *Avg T* the runtime averaged across all instances of the problem class. The last row *# BKS* counts the number of (current) BKS achieved. **Bold** entries state the BKS, entries additionally marked with an asterisk (*) are improving solutions against the previous BKS. Tables B.7 - B.9 detail the solutions of newly found BKS. The new BKS for LRP instance *P122212* was found during parameter tuning. Please note that customer indices start at $n_s + 1$, e.g., instance *P122212* has ten depots, so the first customer is denoted by $n_s + 1 = 11$.

Table B.4: Detailed results on 2E-VRP instances

| Instance | BKS | HALNS | | | | |
|-----------------------------|--------|---------------|---------------|--------------|-------------|-------------|
| | | Best 5 | Avg 5 | Best Gap [%] | Avg Gap [%] | Runtime [s] |
| Set2a_E-n22-k4-s6-17 | 417.07 | 417.07 | 417.07 | 0 | 0 | 25.96 |
| Set2a_E-n22-k4-s8-14 | 384.96 | 384.96 | 384.96 | 0 | 0 | 27.96 |
| Set2a_E-n22-k4-s9-19 | 470.6 | 470.6 | 470.6 | 0 | 0 | 28.03 |
| Set2a_E-n22-k4-s10-14 | 371.5 | 371.5 | 371.5 | 0 | 0 | 26.59 |
| Set2a_E-n22-k4-s11-12 | 427.22 | 427.22 | 427.22 | 0 | 0 | 28.52 |
| Set2a_E-n22-k4-s12-16 | 392.78 | 392.78 | 392.78 | 0 | 0 | 28.41 |
| Set2a_E-n33-k4-s1-9 | 730.16 | 730.16 | 730.16 | 0 | 0 | 37.84 |
| Set2a_E-n33-k4-s2-13 | 714.63 | 714.63 | 714.63 | 0 | 0 | 40.82 |
| Set2a_E-n33-k4-s3-17 | 707.48 | 707.48 | 707.48 | 0 | 0 | 37.06 |
| Set2a_E-n33-k4-s4-5 | 778.74 | 778.74 | 778.74 | 0 | 0 | 39.66 |
| Set2a_E-n33-k4-s7-25 | 756.85 | 756.85 | 756.85 | 0 | 0 | 36.99 |
| Set2a_E-n33-k4-s14-22 | 779.05 | 779.05 | 779.05 | 0 | 0 | 37.08 |
| Set2b_E-n51-k5-s2-17 | 597.49 | 597.49 | 597.49 | 0 | 0 | 62.86 |
| Set2b_E-n51-k5-s4-46 | 530.76 | 530.76 | 530.76 | 0 | 0 | 59.79 |
| Set2b_E-n51-k5-s6-12 | 554.81 | 554.81 | 554.81 | 0 | 0 | 65.6 |
| Set2b_E-n51-k5-s11-19 | 581.64 | 581.64 | 581.64 | 0 | 0 | 61.65 |
| Set2b_E-n51-k5-s27-47 | 538.22 | 538.22 | 538.22 | 0 | 0 | 56.73 |
| Set2b_E-n51-k5-s32-37 | 552.28 | 552.28 | 552.28 | 0 | 0 | 58.8 |
| Set2b_E-n51-k5-s2-4-17-46 | 530.76 | 530.76 | 530.76 | 0 | 0 | 77.42 |
| Set2b_E-n51-k5-s6-12-32-37 | 531.92 | 531.92 | 531.92 | 0 | 0 | 105.35 |
| Set2b_E-n51-k5-s11-19-27-47 | 527.63 | 527.63 | 527.63 | 0 | 0 | 85.39 |
| Set2c_E-n51-k5-s2-17 | 601.39 | 601.39 | 601.39 | 0 | 0 | 58.94 |
| Set2c_E-n51-k5-s4-46 | 702.33 | 702.33 | 702.33 | 0 | 0 | 62.27 |
| Set2c_E-n51-k5-s6-12 | 567.42 | 567.42 | 567.42 | 0 | 0 | 61.74 |
| Set2c_E-n51-k5-s11-19 | 617.42 | 617.42 | 617.42 | 0 | 0 | 64.43 |
| Set2c_E-n51-k5-s27-47 | 530.76 | 530.76 | 530.76 | 0 | 0 | 58.71 |
| Set2c_E-n51-k5-s32-37 | 752.59 | 752.6 | 752.6 | 0 | 0 | 68.19 |
| Set2c_E-n51-k5-s2-4-17-46 | 601.39 | 601.39 | 601.39 | 0 | 0 | 72.71 |
| Set2c_E-n51-k5-s6-12 | 567.42 | 567.42 | 567.42 | 0 | 0 | 61.74 |
| Set2c_E-n51-k5-s11-19-27-47 | 530.76 | 530.76 | 530.76 | 0 | 0 | 70.44 |
| Set3_E-n22-k4-s13-14 | 526.15 | 526.15 | 526.15 | 0 | 0 | 28.38 |
| Set3_E-n22-k4-s13-16 | 521.09 | 521.09 | 521.09 | 0 | 0 | 28.96 |
| Set3_E-n22-k4-s13-17 | 496.38 | 496.38 | 496.38 | 0 | 0 | 26.58 |
| Set3_E-n22-k4-s14-19 | 498.8 | 498.8 | 498.8 | 0 | 0 | 32.09 |
| Set3_E-n22-k4-s17-19 | 512.8 | 512.8 | 512.8 | 0 | 0 | 30.93 |
| Set3_E-n22-k4-s19-21 | 520.42 | 520.42 | 520.42 | 0 | 0 | 29.58 |
| Set3_E-n33-k4-s16-22 | 672.17 | 672.17 | 672.17 | 0 | 0 | 40.96 |
| Set3_E-n33-k4-s16-24 | 666.02 | 666.02 | 666.02 | 0 | 0 | 39.23 |
| Set3_E-n33-k4-s19-26 | 680.36 | 680.37 | 680.37 | 0 | 0 | 37.8 |

Continued on next page

Appendix B Appendix for HALNS for VRPs with Depot Location Decisions

Table B.4 – Continued from previous page

| | | | | | | |
|----------------------|----------------|----------------|----------------|------|------|--------|
| Set3_E-n33-k4-s22-26 | 680.36 | 680.37 | 680.37 | 0 | 0 | 37.94 |
| Set3_E-n33-k4-s24-28 | 670.43 | 670.43 | 670.43 | 0 | 0 | 38.84 |
| Set3_E-n33-k4-s25-28 | 650.58 | 650.58 | 650.58 | 0 | 0 | 38.74 |
| Set3_E-n51-k5-s12-18 | 690.59 | 690.59 | 690.59 | 0 | 0 | 70.25 |
| Set3_E-n51-k5-s12-41 | 683.05 | 683.05 | 683.84 | 0 | 0.12 | 72.84 |
| Set3_E-n51-k5-s12-43 | 710.41 | 710.41 | 710.41 | 0 | 0 | 61.4 |
| Set3_E-n51-k5-s39-41 | 728.54 | 728.54 | 728.54 | 0 | 0 | 68.18 |
| Set3_E-n51-k5-s40-41 | 723.75 | 723.75 | 724.28 | 0 | 0.07 | 98.94 |
| Set3_E-n51-k5-s40-43 | 752.15 | 752.15 | 752.15 | 0 | 0 | 73.59 |
| Set3_E-n51-k5-s13-19 | 560.73 | 560.73 | 560.73 | 0 | 0 | 62.19 |
| Set3_E-n51-k5-s13-42 | 564.45 | 564.45 | 564.45 | 0 | 0 | 61.64 |
| Set3_E-n51-k5-s13-44 | 564.45 | 564.45 | 564.45 | 0 | 0 | 61.49 |
| Set3_E-n51-k5-s40-42 | 746.31 | 746.31 | 746.31 | 0 | 0 | 68.12 |
| Set3_E-n51-k5-s41-42 | 771.56 | 771.56 | 771.56 | 0 | 0 | 67.28 |
| Set3_E-n51-k5-s41-44 | 802.91 | 802.91 | 802.91 | 0 | 0 | 75.83 |
| Set4a_Instance50-1 | 1569.42 | 1569.42 | 1569.42 | 0 | 0 | 59.6 |
| Set4a_Instance50-2 | 1438.33 | 1438.32 | 1438.32 | 0 | 0 | 63.36 |
| Set4a_Instance50-3 | 1570.43 | 1570.43 | 1570.43 | 0 | 0 | 58.52 |
| Set4a_Instance50-4 | 1424.04 | 1424.04 | 1424.04 | 0 | 0 | 57.82 |
| Set4a_Instance50-5 | 2193.52 | 2193.52 | 2193.52 | 0 | 0 | 69.28 |
| Set4a_Instance50-6 | 1279.87 | 1279.89 | 1279.89 | 0 | 0 | 56.05 |
| Set4a_Instance50-7 | 1458.63 | 1458.6 | 1458.6 | 0 | 0 | 54.47 |
| Set4a_Instance50-8 | 1363.74 | 1363.76 | 1363.76 | 0 | 0 | 58.74 |
| Set4a_Instance50-9 | 1450.27 | 1450.25 | 1450.25 | 0 | 0 | 55.81 |
| Set4a_Instance50-10 | 1407.65 | 1407.65 | 1407.65 | 0 | 0 | 56.94 |
| Set4a_Instance50-11 | 2047.46 | 2047.43 | 2047.43 | 0 | 0 | 68.68 |
| Set4a_Instance50-12 | 1209.42 | 1209.46 | 1209.46 | 0 | 0 | 61.3 |
| Set4a_Instance50-13 | 1481.83 | 1481.8 | 1481.8 | 0 | 0 | 66.36 |
| Set4a_Instance50-14 | 1393.61 | 1393.64 | 1393.64 | 0 | 0 | 63.87 |
| Set4a_Instance50-15 | 1489.94 | 1489.92 | 1489.92 | 0 | 0 | 62.26 |
| Set4a_Instance50-16 | 1389.17 | 1389.2 | 1389.2 | 0 | 0 | 58.22 |
| Set4a_Instance50-17 | 2088.49 | 2088.48 | 2088.48 | 0 | 0 | 101.54 |
| Set4a_Instance50-18 | 1227.61 | 1227.68 | 1227.68 | 0.01 | 0.01 | 58.3 |
| Set4a_Instance50-19 | 1564.66 | 1564.66 | 1564.66 | 0 | 0 | 94.04 |
| Set4a_Instance50-20 | 1272.97 | 1272.98 | 1272.98 | 0 | 0 | 68.38 |
| Set4a_Instance50-21 | 1577.82 | 1577.82 | 1577.82 | 0 | 0 | 113.89 |
| Set4a_Instance50-22 | 1281.83 | 1281.83 | 1281.83 | 0 | 0 | 73.3 |
| Set4a_Instance50-23 | 1807.35 | 1807.35 | 1807.35 | 0 | 0 | 146.68 |
| Set4a_Instance50-24 | 1282.68 | 1282.69 | 1282.69 | 0 | 0 | 71.11 |
| Set4a_Instance50-25 | 1522.42 | 1522.4 | 1522.4 | 0 | 0 | 108.17 |
| Set4a_Instance50-26 | 1167.46 | 1167.47 | 1167.47 | 0 | 0 | 64.15 |
| Set4a_Instance50-27 | 1481.57 | 1481.56 | 1481.77 | 0 | 0.01 | 125.24 |
| Set4a_Instance50-28 | 1210.44 | 1210.46 | 1210.46 | 0 | 0 | 71.09 |
| Set4a_Instance50-29 | 1722.04 | 1722 | 1722 | 0 | 0 | 150.95 |
| Set4a_Instance50-30 | 1211.59 | 1211.63 | 1211.63 | 0 | 0 | 62.02 |
| Set4a_Instance50-31 | 1490.33 | 1490.32 | 1490.32 | 0 | 0 | 99.38 |
| Set4a_Instance50-32 | 1199 | 1199.05 | 1199.05 | 0 | 0 | 72.38 |
| Set4a_Instance50-33 | 1508.3 | 1508.32 | 1508.86 | 0 | 0.04 | 123.17 |
| Set4a_Instance50-34 | 1233.92 | 1233.96 | 1233.96 | 0 | 0 | 66.05 |
| Set4a_Instance50-35 | 1718.41 | 1718.42 | 1718.42 | 0 | 0 | 134.33 |
| Set4a_Instance50-36 | 1228.89 | 1228.95 | 1228.95 | 0 | 0 | 65.75 |
| Set4a_Instance50-37 | 1528.73 | 1528.73 | 1528.73 | 0 | 0 | 214.77 |
| Set4a_Instance50-38 | 1169.2 | 1169.2 | 1169.2 | 0 | 0 | 152.93 |
| Set4a_Instance50-39 | 1520.92 | 1520.92 | 1520.92 | 0 | 0 | 190.49 |
| Set4a_Instance50-40 | 1199.42 | 1199.42 | 1199.42 | 0 | 0 | 140.67 |

Continued on next page

Appendix B Appendix for HALNS for VRPs with Depot Location Decisions

Table B.4 – Continued from previous page

| | | | | | | |
|---------------------|---------|----------------|----------------|------|------|--------|
| Set4a_Instance50-41 | 1667.96 | 1667.96 | 1667.96 | 0 | 0 | 259.01 |
| Set4a_Instance50-42 | 1194.54 | 1194.54 | 1194.54 | 0 | 0 | 136.1 |
| Set4a_Instance50-43 | 1439.67 | 1439.67 | 1440.59 | 0 | 0.06 | 209.83 |
| Set4a_Instance50-44 | 1045.13 | 1045.14 | 1045.14 | 0 | 0 | 128.74 |
| Set4a_Instance50-45 | 1450.95 | 1450.95 | 1450.95 | 0 | 0 | 249.41 |
| Set4a_Instance50-46 | 1088.77 | 1088.79 | 1088.79 | 0 | 0 | 143.61 |
| Set4a_Instance50-47 | 1587.29 | 1587.29 | 1587.29 | 0 | 0 | 212.73 |
| Set4a_Instance50-48 | 1082.2 | 1082.21 | 1082.21 | 0 | 0 | 113.16 |
| Set4a_Instance50-49 | 1434.88 | 1434.88 | 1434.88 | 0 | 0 | 183.78 |
| Set4a_Instance50-50 | 1083.16 | 1083.16 | 1083.16 | 0 | 0 | 144.94 |
| Set4a_Instance50-51 | 1398.05 | 1398.03 | 1398.03 | 0 | 0 | 175.27 |
| Set4a_Instance50-52 | 1125.69 | 1125.69 | 1125.69 | 0 | 0 | 95.33 |
| Set4a_Instance50-53 | 1567.77 | 1567.79 | 1568.52 | 0 | 0.05 | 189.32 |
| Set4a_Instance50-54 | 1127.61 | 1127.66 | 1132.03 | 0 | 0.39 | 142.55 |
| Set4b_Instance50-1 | 1569.42 | 1569.42 | 1569.42 | 0 | 0 | 65.73 |
| Set4b_Instance50-2 | 1438.33 | 1438.32 | 1438.32 | 0 | 0 | 74.04 |
| Set4b_Instance50-3 | 1570.34 | 1570.43 | 1570.43 | 0.01 | 0.01 | 66.89 |
| Set4b_Instance50-4 | 1424.04 | 1424.04 | 1424.04 | 0 | 0 | 64.9 |
| Set4b_Instance50-5 | 2193.52 | 2193.52 | 2193.52 | 0 | 0 | 70.7 |
| Set4b_Instance50-6 | 1279.87 | 1279.89 | 1279.89 | 0 | 0 | 63.25 |
| Set4b_Instance50-7 | 1408.57 | 1408.58 | 1408.58 | 0 | 0 | 66.29 |
| Set4b_Instance50-8 | 1360.32 | 1360.32 | 1360.32 | 0 | 0 | 65.19 |
| Set4b_Instance50-9 | 1403.53 | 1403.53 | 1403.53 | 0 | 0 | 60.95 |
| Set4b_Instance50-10 | 1360.56 | 1360.54 | 1360.54 | 0 | 0 | 61.48 |
| Set4b_Instance50-11 | 2047.46 | 2047.43 | 2047.43 | 0 | 0 | 88.55 |
| Set4b_Instance50-12 | 1209.42 | 1209.46 | 1209.46 | 0 | 0 | 64.11 |
| Set4b_Instance50-13 | 1450.93 | 1450.94 | 1450.94 | 0 | 0 | 60.66 |
| Set4b_Instance50-14 | 1393.61 | 1393.64 | 1393.64 | 0 | 0 | 64.43 |
| Set4b_Instance50-15 | 1466.83 | 1466.84 | 1466.84 | 0 | 0 | 62.12 |
| Set4b_Instance50-16 | 1387.83 | 1387.85 | 1387.85 | 0 | 0 | 67.61 |
| Set4b_Instance50-17 | 2088.49 | 2088.48 | 2088.48 | 0 | 0 | 74.91 |
| Set4b_Instance50-18 | 1227.61 | 1227.68 | 1227.68 | 0.01 | 0.01 | 62.27 |
| Set4b_Instance50-19 | 1546.28 | 1546.28 | 1546.91 | 0 | 0.04 | 96.27 |
| Set4b_Instance50-20 | 1272.97 | 1272.98 | 1272.98 | 0 | 0 | 69.84 |
| Set4b_Instance50-21 | 1577.82 | 1577.82 | 1577.82 | 0 | 0 | 97.49 |
| Set4b_Instance50-22 | 1281.83 | 1281.83 | 1281.83 | 0 | 0 | 74.92 |
| Set4b_Instance50-23 | 1652.98 | 1652.98 | 1652.98 | 0 | 0 | 78.49 |
| Set4b_Instance50-24 | 1282.68 | 1282.69 | 1282.69 | 0 | 0 | 71.72 |
| Set4b_Instance50-25 | 1408.57 | 1408.58 | 1408.58 | 0 | 0 | 70.13 |
| Set4b_Instance50-26 | 1167.46 | 1167.47 | 1167.47 | 0 | 0 | 66.26 |
| Set4b_Instance50-27 | 1444.5 | 1444.49 | 1444.49 | 0 | 0 | 105.95 |
| Set4b_Instance50-28 | 1210.44 | 1210.46 | 1210.46 | 0 | 0 | 71.68 |
| Set4b_Instance50-29 | 1552.66 | 1552.66 | 1552.66 | 0 | 0 | 96.53 |
| Set4b_Instance50-30 | 1211.49 | 1211.63 | 1211.63 | 0.01 | 0.01 | 66.07 |
| Set4b_Instance50-31 | 1440.86 | 1440.85 | 1441.01 | 0 | 0.01 | 99.36 |
| Set4b_Instance50-32 | 1199 | 1199.05 | 1199.05 | 0 | 0 | 71.74 |
| Set4b_Instance50-33 | 1478.86 | 1478.87 | 1478.87 | 0 | 0 | 85.78 |
| Set4b_Instance50-34 | 1233.92 | 1233.96 | 1233.96 | 0 | 0 | 65.83 |
| Set4b_Instance50-35 | 1570.72 | 1570.73 | 1570.73 | 0 | 0 | 76.54 |
| Set4b_Instance50-36 | 1228.89 | 1228.95 | 1228.95 | 0 | 0 | 65.35 |
| Set4b_Instance50-37 | 1528.73 | 1528.73 | 1528.73 | 0 | 0 | 162.83 |
| Set4b_Instance50-38 | 1163.07 | 1163.07 | 1163.07 | 0 | 0 | 112.17 |
| Set4b_Instance50-39 | 1520.92 | 1520.92 | 1520.92 | 0 | 0 | 179.79 |
| Set4b_Instance50-40 | 1163.04 | 1163.04 | 1164.76 | 0 | 0.15 | 172.62 |
| Set4b_Instance50-41 | 1652.98 | 1652.98 | 1652.98 | 0 | 0 | 127.34 |

Continued on next page

Appendix B Appendix for HALNS for VRPs with Depot Location Decisions

Table B.4 – Continued from previous page

| | | | | | | |
|---------------------|---------|----------------|----------------|------|------|---------|
| Set4b_Instance50-42 | 1190.17 | 1190.17 | 1190.17 | 0 | 0 | 105.48 |
| Set4b_Instance50-43 | 1406.11 | 1406.1 | 1406.1 | 0 | 0 | 168.45 |
| Set4b_Instance50-44 | 1035.03 | 1035.05 | 1035.05 | 0 | 0 | 103.48 |
| Set4b_Instance50-45 | 1401.87 | 1401.87 | 1401.87 | 0 | 0 | 148.67 |
| Set4b_Instance50-46 | 1058.11 | 1058.1 | 1058.1 | 0 | 0 | 144.73 |
| Set4b_Instance50-47 | 1552.66 | 1552.66 | 1552.66 | 0 | 0 | 132.45 |
| Set4b_Instance50-48 | 1074.5 | 1074.51 | 1074.51 | 0 | 0 | 91.28 |
| Set4b_Instance50-49 | 1434.88 | 1434.88 | 1434.88 | 0 | 0 | 148.69 |
| Set4b_Instance50-50 | 1065.25 | 1065.3 | 1065.3 | 0 | 0 | 87.9 |
| Set4b_Instance50-51 | 1387.51 | 1387.51 | 1387.51 | 0 | 0 | 147.36 |
| Set4b_Instance50-52 | 1103.42 | 1103.47 | 1104.17 | 0 | 0.07 | 121.2 |
| Set4b_Instance50-53 | 1545.73 | 1545.76 | 1545.76 | 0 | 0 | 97.61 |
| Set4b_Instance50-54 | 1113.62 | 1113.66 | 1113.66 | 0 | 0 | 90.87 |
| Set5_100-5-1 | 1564.46 | 1571.43 | 1572.77 | 0.45 | 0.53 | 809.82 |
| Set5_100-5-1b | 1099.35 | 1109.27 | 1109.27 | 0.9 | 0.9 | 377.91 |
| Set5_100-5-2 | 1016.32 | 1016.32 | 1017.99 | 0 | 0.16 | 491.56 |
| Set5_100-5-2b | 782.25 | 783.39 | 783.39 | 0.15 | 0.15 | 297.13 |
| Set5_100-5-3 | 1045.29 | 1045.29 | 1046 | 0 | 0.07 | 589.98 |
| Set5_100-5-3b | 828.54 | 828.54 | 828.54 | 0 | 0 | 279.19 |
| Set5_100-10-1 | 1124.93 | 1124.93 | 1124.93 | 0 | 0 | 2919.96 |
| Set5_100-10-1b | 911.8 | 913.59 | 913.59 | 0.2 | 0.2 | 1258.56 |
| Set5_100-10-2 | 985.4 | 996.3 | 996.74 | 1.11 | 1.15 | 1195.51 |
| Set5_100-10-2b | 766.28 | 768.13 | 768.13 | 0.24 | 0.24 | 750.73 |
| Set5_100-10-3 | 1042.63 | 1042.63 | 1042.63 | 0 | 0 | 1675.63 |
| Set5_100-10-3b | 848.16 | 848.16 | 849.56 | 0 | 0.17 | 1601.16 |
| Set5_200-10-1 | 1537.52 | 1544.1 | 1545.98 | 0.43 | 0.55 | 8035.98 |
| Set5_200-10-1b | 1173.07 | 1175.27 | 1178.51 | 0.19 | 0.46 | 3751.62 |
| Set5_200-10-2 | 1352.87 | 1353.21 | 1353.74 | 0.03 | 0.06 | 2947.85 |
| Set5_200-10-2b | 985.99 | 987.44 | 988.29 | 0.15 | 0.23 | 1635.62 |
| Set5_200-10-3 | 1777.49 | 1782.81 | 1784.61 | 0.3 | 0.4 | 3905.13 |
| Set5_200-10-3b | 1192.35 | 1197.97 | 1198.67 | 0.47 | 0.53 | 2283.3 |
| Set6_A-n51-4 | 652 | 652 | 652 | 0 | 0 | 111.32 |
| Set6_A-n51-5 | 663.41 | 663.41 | 663.41 | 0 | 0 | 136.18 |
| Set6_A-n51-6 | 662.51 | 662.51 | 662.51 | 0 | 0 | 227.35 |
| Set6_B-n51-4 | 563.98 | 563.98 | 563.98 | 0 | 0 | 92.84 |
| Set6_B-n51-5 | 549.23 | 549.23 | 549.23 | 0 | 0 | 113.16 |
| Set6_B-n51-6 | 556.32 | 556.32 | 556.32 | 0 | 0 | 172.88 |
| Set6_C-n51-4 | 689.18 | 689.18 | 689.18 | 0 | 0 | 109.08 |
| Set6_C-n51-5 | 723.12 | 723.12 | 723.12 | 0 | 0 | 108.27 |
| Set6_C-n51-6 | 697 | 697 | 697 | 0 | 0 | 156.1 |
| Set6_A-n76-4 | 985.95 | 985.95 | 986.01 | 0 | 0.01 | 235.59 |
| Set6_A-n76-5 | 979.15 | 979.15 | 979.15 | 0 | 0 | 472.36 |
| Set6_A-n76-6 | 970.2 | 970.2 | 970.2 | 0 | 0 | 612.56 |
| Set6_B-n76-4 | 792.73 | 792.73 | 792.73 | 0 | 0 | 236.2 |
| Set6_B-n76-5 | 783.93 | 783.93 | 784.09 | 0 | 0.02 | 351.29 |
| Set6_B-n76-6 | 774.17 | 774.17 | 775.5 | 0 | 0.17 | 459.42 |
| Set6_C-n76-4 | 1054.89 | 1054.89 | 1054.89 | 0 | 0 | 254.82 |
| Set6_C-n76-5 | 1115.32 | 1115.32 | 1115.32 | 0 | 0 | 377.73 |
| Set6_C-n76-6 | 1060.52 | 1060.52 | 1062.85 | 0 | 0.22 | 745.98 |
| Set6_A-n101-4 | 1194.17 | 1194.17 | 1194.38 | 0 | 0.02 | 326.11 |
| Set6_A-n101-5 | 1211.38 | 1214.41 | 1214.41 | 0.25 | 0.25 | 699.03 |
| Set6_A-n101-6 | 1155.89 | 1155.94 | 1156.56 | 0 | 0.06 | 959.49 |
| Set6_B-n101-4 | 939.21 | 939.83 | 939.83 | 0.07 | 0.07 | 319.94 |
| Set6_B-n101-5 | 967.82 | 969.07 | 969.07 | 0.13 | 0.13 | 763.59 |
| Set6_B-n101-6 | 960.29 | 960.29 | 960.81 | 0 | 0.05 | 688.98 |

Continued on next page

Appendix B Appendix for HALNS for VRPs with Depot Location Decisions

Table B.4 – Continued from previous page

| | | | | | | |
|----------------------|---------|----------------|---------|---------|--------|--------|
| Set6_C-n101-4 | 1292.04 | 1292.04 | 1297.26 | 0 | 0.4 | 509.41 |
| Set6_C-n101-5 | 1304.86 | 1305.82 | 1305.82 | 0.07 | 0.07 | 552.44 |
| Set6_C-n101-6 | 1284.48 | 1284.48 | 1290.35 | 0 | 0.46 | 796.4 |
| Avg. gap / avg. time | | | 0.00% | 0.025% | 0.043% | 285.8s |
| # BKS | 207/207 | | | 190/207 | | |

Table B.5: Detailed results on LRP instances

| Instance | BKS | HALNS | | | | Runtime [s] |
|------------|---------|----------------|----------------|--------------|-------------|-------------|
| | | Best 10 | Avg 10 | Best Gap [%] | Avg Gap [%] | |
| 20-5-1a | 54793 | 54793 | 54793 | 0 | 0 | 25.61 |
| 20-5-1b | 39104 | 39104 | 39104 | 0 | 0 | 21.3 |
| 20-5-2a | 48908 | 48908 | 48908 | 0 | 0 | 23.1 |
| 20-5-2b | 37542 | 37542 | 37542 | 0 | 0 | 22.61 |
| 50-5-1 | 90111 | 90111 | 90111 | 0 | 0 | 61.47 |
| 50-5-1b | 63242 | 63242 | 63242 | 0 | 0 | 54.47 |
| 50-5-2 | 88298 | 88298 | 88298 | 0 | 0 | 59.08 |
| 50-5-2b | 67308 | 67308 | 67308 | 0 | 0 | 58.93 |
| 50-5-2bis | 84055 | 84055 | 84055 | 0 | 0 | 61.26 |
| 50-5-2bbis | 51822 | 51822 | 51822 | 0 | 0 | 52.94 |
| 50-5-3 | 86203 | 86203 | 86203 | 0 | 0 | 66.52 |
| 50-5-3b | 61830 | 61830 | 61830 | 0 | 0 | 54.99 |
| 100-5-1 | 274814 | 275079 | 275120.6 | 0.1 | 0.11 | 385.09 |
| 100-5-1b | 213568 | 213568 | 213588.6 | 0 | 0.01 | 279.77 |
| 100-5-2 | 193671 | 193671 | 193671 | 0 | 0 | 210.24 |
| 100-5-2b | 157095 | 157095 | 157095 | 0 | 0 | 186.1 |
| 100-5-3 | 200079 | 200079 | 200079 | 0 | 0 | 247.28 |
| 100-5-3b | 152441 | 152441 | 152441 | 0 | 0 | 138.93 |
| 100-10-1 | 287661 | 287692 | 287870 | 0.01 | 0.07 | 394.79 |
| 100-10-1b | 230989 | 230989 | 230989 | 0 | 0 | 222.9 |
| 100-10-2 | 243590 | 243590 | 243611 | 0 | 0.01 | 254.71 |
| 100-10-2b | 203988 | 203988 | 203988 | 0 | 0 | 146.75 |
| 100-10-3 | 250882 | 250882 | 250945.6 | 0 | 0.03 | 332.39 |
| 100-10-3b | 203114 | 203114 | 203404.6 | 0 | 0.14 | 207.38 |
| 200-10-1 | 474850 | 476472 | 477174.2 | 0.34 | 0.49 | 1478.59 |
| 200-10-1b | 375177 | 375346 | 375512.6 | 0.05 | 0.09 | 1031.6 |
| 200-10-2 | 448077 | 448721 | 449138.6 | 0.14 | 0.24 | 1283.05 |
| 200-10-2b | 373696 | 373696 | 373706.2 | 0 | 0 | 951.11 |
| 200-10-3 | 469433 | 470422 | 471149 | 0.21 | 0.37 | 1197.06 |
| 200-10-3b | 362320 | 362630 | 362744.4 | 0.09 | 0.12 | 1091.56 |
| P111112 | 1467.68 | 1467.68 | 1467.68 | 0 | 0 | 163.22 |
| P111122 | 1448.37 | 1448.37 | 1448.54 | 0 | 0.01 | 239.21 |
| P111212 | 1394.8 | 1394.8 | 1394.8 | 0 | 0 | 162.94 |
| P111222 | 1432.29 | 1432.29 | 1432.6 | 0 | 0.02 | 234.86 |
| P112112 | 1167.16 | 1167.16 | 1167.16 | 0 | 0 | 184.35 |
| P112122 | 1102.24 | 1102.24 | 1102.24 | 0 | 0 | 151.51 |
| P112212 | 791.66 | 791.66 | 791.66 | 0 | 0 | 168.88 |
| P112222 | 728.3 | 728.3 | 728.3 | 0 | 0 | 137.87 |
| P113112 | 1238.24 | 1238.49 | 1238.49 | 0.02 | 0.02 | 197.3 |
| P113122 | 1245.3 | 1245.31 | 1245.46 | 0 | 0.01 | 150.63 |
| P113212 | 902.26 | 902.26 | 902.26 | 0 | 0 | 141.88 |
| P113222 | 1018.29 | 1018.29 | 1020.09 | 0 | 0.18 | 154.53 |
| P131112 | 1892.17 | 1892.17 | 1893.86 | 0 | 0.09 | 481.69 |

Continued on next page

Appendix B Appendix for HALNS for VRPs with Depot Location Decisions

Table B.5 – Continued from previous page

| | | | | | | |
|----------------------|---------|----------------|----------------|---------------|--------|---------|
| P131122 | 1819.68 | 1819.68 | 1822.7 | 0 | 0.17 | 473.25 |
| P131212 | 1960.02 | 1964.34 | 1965.4 | 0.22 | 0.27 | 406.93 |
| P131222 | 1792.77 | 1792.77 | 1795.23 | 0 | 0.14 | 377.68 |
| P132112 | 1443.32 | 1443.32 | 1443.32 | 0 | 0 | 368.65 |
| P132122 | 1429.3 | 1429.3 | 1439 | 0 | 0.68 | 441.85 |
| P132212 | 1204.42 | 1204.42 | 1204.42 | 0 | 0 | 427.53 |
| P132222 | 924.68 | 924.68 | 924.68 | 0 | 0 | 484.31 |
| P133112 | 1694.18 | 1694.68 | 1698.6 | 0.03 | 0.26 | 530.37 |
| P133122 | 1392.01 | 1392.01 | 1393.74 | 0 | 0.12 | 473.67 |
| P133212 | 1197.95 | 1197.95 | 1197.95 | 0 | 0 | 478.82 |
| P133222 | 1151.37 | 1151.69 | 1151.78 | 0.03 | 0.04 | 385.31 |
| P121112 | 2237.73 | 2238.59 | 2245.18 | 0.04 | 0.33 | 856.79 |
| P121122 | 2137.45 | 2137.45 | 2140.31 | 0 | 0.13 | 1011.3 |
| P121212 | 2195.17 | 2195.17 | 2199.6 | 0 | 0.2 | 858.64 |
| P121222 | 2214.86 | 2214.86 | 2216.2 | 0 | 0.06 | 1014.78 |
| P122112 | 2070.43 | 2070.43 | 2072.01 | 0 | 0.08 | 861.55 |
| P122122 | 1685.52 | 1685.52 | 1685.69 | 0 | 0.01 | 1029.48 |
| P122212 | 1449.93 | 1449.62 | 1449.77 | -0.02 | -0.01 | 1067.59 |
| P122222 | 1082.46 | 1082.59 | 1082.59 | 0.01 | 0.01 | 842.09 |
| P123112 | 1942.23 | 1949.95 | 1952.82 | 0.4 | 0.55 | 1047.41 |
| P123122 | 1910.08 | 1910.08 | 1913.17 | 0 | 0.16 | 1005.32 |
| P123212 | 1761.11 | 1760.2 | 1761.14 | -0.05 | 0 | 1118.7 |
| P123222 | 1390.86 | 1390.74 | 1390.79 | -0.01 | -0.01 | 663.32 |
| Christ50 | 565.6 | 565.6 | 565.6 | 0 | 0 | 54.96 |
| Christ75 | 848.85 | 848.85 | 848.85 | 0 | 0 | 120.92 |
| Christ100 | 833.4 | 833.43 | 833.95 | 0 | 0.07 | 196.76 |
| Das88 | 355.78 | 355.78 | 355.78 | 0 | 0 | 91.74 |
| Das150 | 43919.9 | 43919.9 | 43920.6 | 0 | 0 | 410.93 |
| Gaspelle | 424.9 | 424.9 | 424.9 | 0 | 0 | 24.9 |
| Gaspelle2 | 585.11 | 585.11 | 585.11 | 0 | 0 | 23.67 |
| Gaspelle3 | 512.1 | 512.1 | 512.1 | 0 | 0 | 29.05 |
| Gaspelle4 | 562.22 | 562.22 | 562.22 | 0 | 0 | 34.49 |
| Gaspelle5 | 504.33 | 504.33 | 504.33 | 0 | 0 | 32.52 |
| Gaspelle6 | 460.37 | 460.37 | 460.37 | 0 | 0 | 35.88 |
| Min27 | 3062 | 3062.02 | 3062.02 | 0 | 0 | 28.55 |
| Min134 | 5709 | 5709 | 5709 | 0 | 0 | 240.03 |
| Avg. gap / avg. time | | | 0.00% | 0.020% | 0.067% | 388.9s |
| # BKS | 76/79 | | | 62 (3 new)/79 | | |

Table B.6: Detailed results on MDVRP instances

| Instance | BKS | HALNS | | | | Runtime [s] |
|----------------------|---------|----------------|----------------|--------------|-------------|-------------|
| | | Best 10 | Avg 10 | Best Gap [%] | Avg Gap [%] | |
| p01 | 576.87 | 576.87 | 576.87 | 0 | 0 | 60.02 |
| p02 | 473.53 | 473.53 | 473.53 | 0 | 0 | 54.47 |
| p03 | 641.19 | 641.19 | 641.19 | 0 | 0 | 103 |
| p04 | 1001.04 | 1001.04 | 1001.04 | 0 | 0 | 198.66 |
| p05 | 750.03 | 750.03 | 750.03 | 0 | 0 | 145.98 |
| p06 | 876.5 | 876.5 | 876.62 | 0 | 0.01 | 206.59 |
| p07 | 881.97 | 881.97 | 881.97 | 0 | 0 | 203.58 |
| p08 | 4369.95 | 4375.49 | 4379.97 | 0.13 | 0.23 | 1599.68 |
| p09 | 3858.66 | 3862.16 | 3865.39 | 0.09 | 0.17 | 1401.68 |
| p10 | 3629.6 | 3631.37 | 3632.57 | 0.05 | 0.08 | 1653.99 |
| p11 | 3545.18 | 3546.06 | 3546.06 | 0.02 | 0.02 | 1032.74 |
| p12 | 1318.95 | 1318.95 | 1318.95 | 0 | 0 | 100.22 |
| p13 | 1318.95 | 1318.95 | 1318.95 | 0 | 0 | 85.05 |
| p14 | 1360.12 | 1360.12 | 1360.12 | 0 | 0 | 84.2 |
| p15 | 2505.42 | 2505.42 | 2505.42 | 0 | 0 | 334.53 |
| p16 | 2572.23 | 2572.23 | 2572.23 | 0 | 0 | 252.51 |
| p17 | 2709.09 | 2709.09 | 2709.09 | 0 | 0 | 239.79 |
| p18 | 3702.85 | 3702.85 | 3702.85 | 0 | 0 | 1243.84 |
| p19 | 3827.06 | 3827.06 | 3827.06 | 0 | 0 | 538.79 |
| p20 | 4058.07 | 4058.07 | 4058.07 | 0 | 0 | 477.34 |
| p21 | 5474.84 | 5474.84 | 5480.95 | 0 | 0.11 | 3310.95 |
| p22 | 5702.16 | 5702.16 | 5702.16 | 0 | 0 | 1309.81 |
| p23 | 6078.75 | 6078.75 | 6078.75 | 0 | 0 | 1072.26 |
| pr01 | 861.32 | 861.32 | 861.32 | 0 | 0 | 44.35 |
| pr02 | 1307.34 | 1307.34 | 1307.34 | 0 | 0 | 124.84 |
| pr03 | 1803.8 | 1803.8 | 1803.8 | 0 | 0 | 246.15 |
| pr04 | 2058.31 | 2058.31 | 2058.82 | 0 | 0.02 | 578.52 |
| pr05 | 2331.2 | 2331.2 | 2336.91 | 0 | 0.24 | 1174.63 |
| pr06 | 2676.3 | 2677.8 | 2683.72 | 0.06 | 0.28 | 1709.75 |
| pr07 | 1089.56 | 1089.56 | 1089.56 | 0 | 0 | 69.25 |
| pr08 | 1664.85 | 1664.85 | 1664.85 | 0 | 0 | 338.88 |
| pr09 | 2133.2 | 2133.2 | 2136.49 | 0 | 0.15 | 726.5 |
| pr10 | 2867.26 | 2870.63 | 2879.33 | 0.12 | 0.42 | 2207.43 |
| Avg. gap / avg. time | 0.00% | | | 0.014% | 0.053% | 694.8s |
| # BKS | 33/33 | | | 27/33 | | |

Table B.7: Solution LRP instance *P122212*

| LRP - P122212 BKS 1449.03 | | | | |
|---------------------------------|----------------|----------|----------|---|
| Depot 8 | | | | |
| | Demand 1928 | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 148 | 63.3482 | 63.3482 | 8-101-97-93-153-98-111-123-146-113-128-8 |
| Route 2 | 148 | 37.6835 | 37.6835 | 8-105-94-119-152-100-129-140-137-136-8 |
| Route 3 | 149 | 78.4976 | 78.4976 | 8-186-170-207-180-167-187-182-203-209-175-188-8 |
| Route 4 | 148 | 105.137 | 105.137 | 8-99-104-106-11-174-194-190-178-208-164-8 |
| Route 5 | 149 | 78.9653 | 78.9653 | 8-184-158-179-199-172-196-161-173-210-189-191-8 |
| Route 6 | 147 | 55.6065 | 55.6065 | 8-92-121-122-147-88-133-86-116-91-115-8 |
| Route 7 | 150 | 45.8852 | 45.8852 | 8-84-109-87-103-142-120-143-131-124-102-8 |
| Route 8 | 150 | 103.39 | 103.39 | 8-165-155-162-198-202-176-157-156-16-201-8 |
| Route 9 | 150 | 49.8417 | 49.8417 | 8-83-117-114-125-110-145-150-138-96-144-8 |
| Route 10 | 149 | 34.1428 | 34.1428 | 8-95-118-130-148-107-89-149-132-90-139-8 |
| Route 11 | 150 | 65.1197 | 65.1197 | 8-112-15-13-108-85-126-151-141-135-134-127-8 |
| Route 12 | 145 | 59.4967 | 59.4967 | 8-168-206-197-183-171-195-166-192-193-205-8 |
| Route 13 | 145 | 63.3845 | 63.3845 | 8-163-200-154-177-185-159-204-160-181-169-8 |
| Depot 9 | | | | |
| | Demand 1009 | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 141 | 22.6563 | 22.6563 | 9-26-22-50-56-21-51-72-58-44-9 |
| Route 2 | 150 | 69.7896 | 69.7896 | 9-36-12-78-14-53-81-68-24-45-77-40-9 |
| Route 3 | 138 | 25.9581 | 25.9581 | 9-41-17-61-76-71-59-33-64-67-9 |
| Route 4 | 149 | 27.1209 | 27.1209 | 9-57-35-20-74-75-47-28-34-60-79-9 |
| Route 5 | 137 | 19.9485 | 19.9485 | 9-18-55-80-42-19-38-25-43-49-73-9 |
| Route 6 | 148 | 18.6211 | 18.6211 | 9-30-31-82-23-54-52-48-37-66-9 |
| Route 7 | 146 | 24.4349 | 24.4349 | 9-65-27-69-39-62-29-32-46-63-70-9 |

Table B.8: Solution LRP instance *P123212*

| LRP - P123212 | | | | |
|----------------|--------|----------------|----------|---|
| BKS | | 1760.20 | | |
| Depot 3 | | | | |
| Demand | | | | |
| 1827 | | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 142 | 82.4028 | 82.4028 | 3-90-60-77-65-84-78-81-88-95-3 |
| Route 2 | 132 | 67.7368 | 67.7368 | 3-16-17-24-50-28-23-27-44-3 |
| Route 3 | 150 | 85.9019 | 85.9019 | 3-20-40-36-42-14-48-46-29-30-52-3 |
| Route 4 | 150 | 72.8714 | 72.8714 | 3-11-43-18-19-37-13-15-45-49-26-3 |
| Route 5 | 147 | 77.7983 | 77.7983 | 3-63-66-69-89-79-93-72-74-57-76-3 |
| Route 6 | 149 | 68.96 | 68.96 | 3-71-82-70-68-80-75-59-92-61-94-3 |
| Route 7 | 145 | 69.8052 | 69.8052 | 3-56-55-96-73-87-58-86-54-91-3 |
| Route 8 | 142 | 56.1821 | 56.1821 | 3-186-198-183-196-188-177-199-190-182-197-3 |
| Route 9 | 148 | 81.946 | 81.946 | 3-33-25-32-21-12-51-62-64-67-85-83-3 |
| Route 10 | 137 | 45.8671 | 45.8671 | 3-178-184-185-205-193-207-180-191-209-3 |
| Route 11 | 149 | 81.2453 | 81.2453 | 3-35-22-34-41-31-53-38-39-47-3 |
| Route 12 | 147 | 63.2245 | 63.2245 | 3-175-201-202-194-187-206-200-204-189-192-203-3 |
| Route 13 | 89 | 43.0705 | 43.0705 | 3-176-210-179-195-181-208-3 |
| Depot 8 | | | | |
| Demand | | | | |
| 1191 | | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 150 | 47.1888 | 47.1888 | 8-107-138-132-126-100-133-121-119-128-8 |
| Route 2 | 147 | 74.5361 | 74.5361 | 8-166-147-159-154-173-152-153-142-171-174-8 |
| Route 3 | 148 | 65.6967 | 65.6967 | 8-151-146-169-162-157-148-163-149-150-155-8 |
| Route 4 | 150 | 49.3432 | 49.3432 | 8-106-97-124-139-135-136-134-103-114-8 |
| Route 5 | 149 | 32.2056 | 32.2056 | 8-125-122-108-102-130-98-116-112-137-8 |
| Route 6 | 149 | 35.6204 | 35.6204 | 8-127-110-113-141-140-118-99-123-120-131-8 |
| Route 7 | 150 | 68.8255 | 68.8255 | 8-144-143-172-156-168-145-164-167-160-158-165-8 |
| Route 8 | 148 | 79.7698 | 79.7698 | 8-129-104-111-115-117-105-101-109-170-161-8 |

Table B.9: Solution LRP instance *P123222*

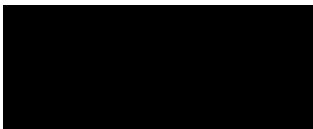
| LRP - P123222 | BKS | 1390.74 | | |
|-----------------|--------|----------|----------|---|
| Depot 3 | Demand | | | |
| | 594 | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 149 | 50.2523 | 50.2523 | 3-105-110-140-119-136-129-137-139-133-128-3 |
| Route 2 | 149 | 44.5297 | 44.5297 | 3-104-122-135-115-113-111-121-142-114-123-3 |
| Route 3 | 148 | 29.2728 | 29.2728 | 3-109-118-107-124-130-131-117-108-116-126-3 |
| Route 4 | 148 | 34.4071 | 34.4071 | 3-106-132-141-125-127-138-112-134-120-3 |
| Depot 4 | Demand | | | |
| | 500 | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 78 | 10.1649 | 10.1649 | 4-204-203-208-202-206-4 |
| Route 2 | 139 | 29.1737 | 29.1737 | 4-193-196-195-209-218-214-188-194-213-4 |
| Route 3 | 139 | 28.4829 | 28.4829 | 4-191-220-207-189-211-210-212-200-197-4 |
| Route 4 | 144 | 24.9624 | 24.9624 | 4-215-219-198-216-201-205-190-192-199-217-4 |
| Depot 13 | Demand | | | |
| | 591 | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 146 | 19.0509 | 19.0509 | 13-77-81-100-89-80-68-98-85-86-13 |
| Route 2 | 150 | 46.464 | 46.464 | 13-88-102-69-97-87-72-91-103-90-94-96-13 |
| Route 3 | 150 | 24.6196 | 24.6196 | 13-70-67-82-71-73-76-78-84-93-13 |
| Route 4 | 145 | 40.7973 | 40.7973 | 13-65-79-99-74-75-101-83-66-92-95-13 |
| Depot 19 | Demand | | | |
| | 690 | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 147 | 44.2391 | 44.2391 | 19-152-186-153-146-180-160-148-167-184-183-19 |
| Route 2 | 137 | 27.384 | 27.384 | 19-157-176-156-155-168-158-159-165-19 |
| Route 3 | 140 | 33.505 | 33.505 | 19-162-182-164-178-154-161-171-173-185-19 |
| Route 4 | 148 | 46.706 | 46.706 | 19-144-175-187-163-145-170-172-147-174-151-19 |
| Route 5 | 118 | 30.0726 | 30.0726 | 19-166-179-149-150-143-169-177-181-19 |
| Depot 20 | Demand | | | |
| | 683 | | | |
| Route | Demand | Distance | Duration | Sequence |
| Route 1 | 150 | 27.9864 | 27.9864 | 20-61-24-34-40-46-25-52-43-63-62-64-20 |
| Route 2 | 143 | 26.132 | 26.132 | 20-22-47-36-26-54-35-32-53-33-48-20 |
| Route 3 | 137 | 18.2386 | 18.2386 | 20-39-57-50-59-21-27-49-41-20 |
| Route 4 | 138 | 21.8631 | 21.8631 | 20-51-30-31-29-28-44-42-55-20 |
| Route 5 | 115 | 12.4397 | 12.4397 | 20-23-56-37-45-58-38-60-20 |

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die schriftliche Dissertationsleistung selbständig und ohne unerlaubte fremde Hilfe angefertigt habe. Ich habe keine anderen als die in der Arbeit angegebenen Schriften und Hilfsmittel benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen kenntlich gemacht. Insbesondere versichere ich, dass ich nicht die Hilfe von Vermittlungs- oder Beratungsdiensten (Promotionsberater oder Promotionsberaterinnen oder andere Personen) in Anspruch genommen habe.

Weiterhin versichere ich, dass ich keine früheren Promotionsversuche unternommen, Promotionen abgeschlossen oder die Dissertation in gleicher oder anderer Form in einem anderen Versuch oder in einem anderen Prüfungsverfahren vorgelegt habe.

Gaimersheim, 30.04.2022



(Markus Frank)