

# Vergleich von Methoden zur Lösung des Capacitated Vehicle Routing Problems (CVRP)

Walter Hoos, Alexander Hübler, David Winzer

Department Sichere Informationssysteme, FH Oberösterreich, Campus Hagenberg

**Abstract**—Das Capacitated Vehicle Routing Problem (CVRP) ist ein kombinatorisches Optimierungsproblem mit vielzähligen praktischen Anwendungen in der Logistik und Tourenplanung. Ziel ist es, eine kostenminimale Reihe von Routen für eine Flotte von Lieferfahrzeugen mit begrenzter Kapazität zu finden, sodass alle Kundennachfragen bedient werden. Das CVRP ist NP-komplex und gehört zu den am intensivsten erforschten Problemen im Bereich der Tourenplanung, wodurch es eine Vielzahl von Lösungsansätzen gibt. Einige von ihnen sollen in dieser Arbeit miteinander verglichen und evaluiert werden.

## I. BESCHREIBUNG DES PROBLEMS

Das Vehicle Routing Problem (VRP), auch bekannt als das Standardproblem der Tourenplanung, repräsentiert ein zentrales kombinatorisches Optimierungsproblem innerhalb der logistischen Planung und des Operations Research. Die grundlegende Zielsetzung des VRP besteht darin, für eine Flotte von Fahrzeugen die optimale Routenplanung zu ermitteln, um eine Menge von Kundenstandorten unter Berücksichtigung bestimmter Restriktionen zu beliefern. Diese Restriktionen können unter anderem die Kapazität der Fahrzeuge, Zeitfenster für die Lieferungen und die maximale Dauer der Touren umfassen.

Konzeptuell stellt das VRP eine Erweiterung des Traveling Salesman Problem (TSP) dar. Während das TSP die Suche nach der kürzesten, möglichen Route für eine/n einzelne/n Verkäufer\*in, der jede Stadt genau einmal besucht und zum Ausgangspunkt zurückkehrt, zum Gegenstand hat, erweitert das VRP diese Problemstellung auf mehrere Fahrzeuge und berücksichtigt zusätzlich eine Reihe von praxisrelevanten Nebenbedingungen. Das VRP trägt somit einer komplexeren und realitätsnäheren Anforderung der Tourenplanung Rechnung. Beschrieben wurde dieses Problem erstmals durch G. B. Dantzig und J. H. Ramser im Jahr 1959.

Im Zuge dieser Arbeit soll eine spezifische Instanz des CVRP welche von P. Augerat[3] beschrieben wurde genutzt werden, um die Effektivität mehrerer Methoden zur Lösung des Problems zu evaluieren.

### A. A-n80-k10

Bei der Problem Instanz handelt es sich konkret um das Problem A-n80-k10 aus dem Set A<sup>1</sup>.

<sup>1</sup><http://www.vrp-rep.org/datasets/download/augerat-1995-set-a.zip>

### B. Formale Beschreibung des Problems

Die formale Beschreibung des Problems wurde mithilfe von [6] durchgeführt.

Der Lösungsraum besteht aus  $K$  Kanten zwischen  $S$  Standorten. Diese Standorte haben jeweils einen Bedarf  $B$ . Eine Strecke zwischen zwei Standorten kann durch ihre zwei Standorte  $(a, b)$  ausgedrückt werden. Es existieren Fahrzeuge  $f$  aus der Menge aller Fahrzeuge  $F$ , welche jeweils eine Maximalladung  $M$  mit sich führen können. Zusätzlich muss eine Binärvariable  $x_{abf}$  eingeführt werden, welche besagt, ob ein Fahrzeug die Strecke  $(a, b)$  nutzt ( $f = 1$ ) oder nicht ( $f = 0$ ). Zudem besagt  $y_{af}$  ob ein Fahrzeug  $f$  den Standort  $a$  besucht ( $f = 1$ ) oder nicht ( $f = 0$ ). Die Kosten der Strecke  $(a, b)$  wird als  $k_{ab}$  beschrieben. Es gilt die Zielfunktion  $Z$  zu minimieren:

$$Z = \sum_{(a,b) \in S} \sum_{f=1}^F x_{abf} k_{ab}$$

Zudem gelten folgende Nebenbedingungen:

- Jeder Standort  $S$  darf nur einmal besucht werden ausgenommen das Depot  $s_0$  :

$$\sum_{f=1}^F y_{af} = 1 \forall a \in S \setminus (s_0)$$

- Jede Route muss am Depot  $s_0$  starten:

$$\sum_{f=1}^F y_{s_0 f} = 1$$

- Ein Standort darf nur genau dann besucht werden wenn  $y_{abf} = 1$ :

$$\sum_b x_{abf} \iff y_{af} = 1 \forall (a, b) \in S$$

- Die Maximalladung darf in keinem der Fahrzeuge überschritten werden:

$$\sum_{a=1}^S B_a y_{af} \leq M \forall f \in S$$

## II. BESCHREIBUNG DER VERWENDETEN WERKZEUGE (METHODIK)

Zur Lösung des CVRP gibt es verschiedene Methoden und Verfahren. Einige der bekanntesten Methoden gehören zur Kategorie der heuristischen Algorithmen, wobei diese sich in zwei Hauptgruppen unterteilen lassen:

TABLE I  
GENUTZTE PARAMETER FÜR OSGA

Parameter	Wert
Elites	5
Crossover	Alba Permutation Crossover
Maximum Generations	10000
Maximum Selection Pressure	100
Mutation Probability	10%
Mutator	AlbaCustomerInsertionManipulator
Population Size	500
Selected Parents	300
Selektor	Tournament Selektor (Group Size 2)

1. Konstruktive Heuristiken: Diese Algorithmen erstellen iterativ eine zulässige Lösung, indem sie schrittweise Kunden zu Routen hinzufügen. Gängige Konstruktionsheuristiken für das CVRP sind beispielsweise die Savings-Heuristik, die Sweep-Heuristik und die Cluster-First, Route-Second-Methode.

2. Verbesserungsheuristiken: Diese Heuristiken nehmen eine bereits existierende zulässige Lösung und versuchen, diese iterativ durch lokale Suchoperationen zu verbessern. Bekannte Verbesserungsheuristiken für das CVRP sind unter anderem die  $\lambda$ -Interchange-Heuristik, die Or-Opt-Heuristik und die Tabu-Suche.

#### A. Offspring Selection Genetic Algorithm (OSGA)

Genetische Algorithmen sind eine Form von metaheuristischen Verfahren, welche die natürliche Selektion in der Genetik nachahmen. Hierbei werden Individuen nach dem Prinzip "survival of the fittest" ausgewählt und so soll über mehrere Generationen hinweg so eine nahezu optimale Lösung generiert werden. Die Grundsätzliche Funktionsweise von GA ist wie folgt [7]:

- 1) Initialisierung einer Population von Chromosomen
- 2) Fitness evaluation der Chromosomen
- 3) Selektion der Eltern
- 4) Variation durch Crossover und Mutation
- 5) Fitness evaluation
- 6) (Überlebensselektion)
- 7) Wenn Abbruchkriterien erfüllt sind Ende, ansonsten Sprung zu 3.

Traditionelle GA haben einen primären Selektionsprozess zum Finden der Eltern. In OSGA existiert neben diesem ein weiterer Selektionsprozess, welcher die Auswahl der Nachkommen steuert [2]. Die genutzte OSGA Instanz wurde mittels der Software Heuristiclab<sup>2</sup> erzeugt. Die genutzten Parameter können in Tabelle I gefunden werden.

OSGA wurde in dieser Konfiguration 20 mal ausgeführt und eine Evaluierung der Ergebnisse erstellt.

In dieser Konfiguration liefert OSGA im besten Fall Ergebnisse mit einer Routenlänge von 3339m bis zu. Die Streuung

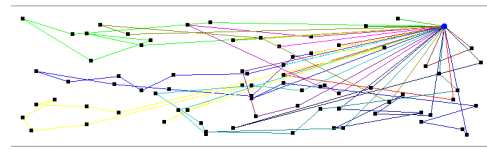


Fig. 1. Beste durch OSGA gefundene Routen

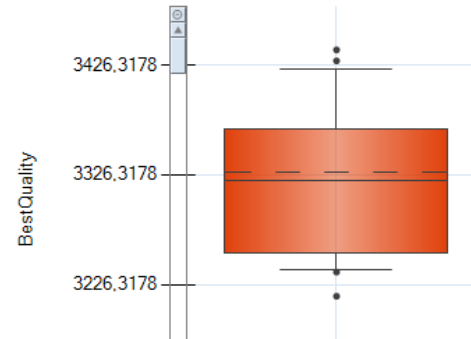


Fig. 2. Streuung der durch OSGA erzielten Ergebnisse

der Ergebnisse ist nicht besonders groß jedoch liegt das best erzielte Ergebnis weit von dem bekannten Optimum. Es ist durchaus möglich, dass durch eine Modifikation einiger Parameter das Ergebnis noch verbessert werden könnte, jedoch ist vermutlich eine andere Metaheuristik besser geeignet um ein gutes Ergebnis zu finden. Ein möglicher Algorithmus hierfür wären beispielsweise Simulated Annealing.

#### B. Google OR-Tools

Die Google OR-Tools [1] sind eine Open-Source-Software-Suite für Operations Research (OR), die von Google entwickelt und bereitgestellt wird. Sie bietet eine Reihe von Algorithmen und Modellierungshilfen für verschiedene Optimierungsprobleme, darunter auch Lösungsansätze für das Capacitated Vehicle Routing Problem (CVRP).

Die OR-Tools sind in mehreren Programmiersprachen wie C++, Python und Java verfügbar und ermöglichen so eine unkomplizierte Integration in bestehende Softwaresysteme. Zudem bieten sie leistungsfähige Visualisierungsmöglichkeiten zur Darstellung der berechneten Routen.

Unsere Implementierung des CVRP in den Google OR-Tools erfolgt durch die Verwendung des Routing-Moduls [4] in Python. Dieses Modul bietet verschiedene Algorithmen und Heuristiken, die speziell für Routing-Probleme entwickelt wurden, sowie weitere Einstellmöglichkeiten für den Routing-Solver wie zum Beispiel:

- `solution_limit`: Maximale Anzahl von Lösungen, die der Solver generieren soll.
- `time_limit.seconds`: Maximale Zeit in Sekunden, die der Solver für die Lösung des Problems verwenden soll.
- `lns_time_limit.seconds`: Maximale Zeit in Sekunden, die der Solver für die Lösung des Problems mit Large Neighborhood Search (LNS) verwenden soll.

<sup>2</sup><https://dev.heuristiclab.com/trac.fcgi/>

Außerdem stehen verschiedene Strategien zur Auswahl, um eine initiale Lösung zu finden wie z.B. kostengünstigste Verbindung, Savings-Algorithmus, Sweep-Algorithmus oder eine Variante des Christofides-Algorithmus. In unserer Implementierung verwenden wir, wie von Google empfohlen [4], die Strategie der kostengünstigsten Verbindung.

Für die lokale Suche zur Verbesserung der initialen Lösung gibt es ebenfalls mehrere metaheuristische Ansätze wie GreedyDescent, GuidedLocalSearch, SimulatedAnnealing oder TabuSearch. Von Google empfohlen und daher von uns verwendet wird die GuidedLocalSearch-Heuristik.

Weitere Parameter erlauben die Steuerung der Propagierung von Constraints und entscheiden, ob vollständige oder nur eingeschränkte Propagierung verwendet wird.

### C. Z3 SMT Solver

Der Z3 Solver ist ein leistungsfähiges Satisfiability-Modulo-Theories (SMT) Werkzeug, das von Microsoft Research entwickelt wurde. Es findet breite Anwendung in verschiedenen Bereichen, wie der formalen Verifikation, der symbolischen Ausführung und der Constraint-Programmierung. Im Kontext des Capacitated Vehicle Routing Problems (CVRP) wird der Z3 Solver als Optimierungsmodul eingesetzt, um effiziente Lösungen für dieses Problem zu finden.

Bei dem CVRP handelt es sich um ein NP-Komplettes Problem, da es zur Klasse NP zählt (nichtdeterministisch polynomiell zeitlösbar). Ein Problem wird als NP-komplett klassifiziert, wenn es zwei Hauptkriterien erfüllt:

- Erstens, jede vorgeschlagene Lösung des Problems kann in polynomieller Zeit verifiziert, jedoch nur in Exponentieller oder Faktorieller Zeit berechnet werden.
- Es kann jedes andere Problem in NP kann in polynomieller Zeit auf dieses Problem reduziert werden. Dies wird üblicherweise durchgeführt, indem ein bekanntes NP-komplettes Problem auf das zu untersuchende Problem reduziert wird.

Da das CVRP eine Verallgemeinerung des TSP ist und das TSP bekanntlich NP-komplett ist, gilt das CVRP ebenfalls als NP-komplett. Dies bedeutet, dass es unwahrscheinlich ist, dass ein Algorithmus existiert, der alle CVRP-Instanzen in polynomieller Zeit lösen kann.

Der Z3-Solver kann prinzipiell auch für NP-komplette Probleme wie das CVRP eingesetzt werden. Allerdings bedeutet die NP-Komplettheit, dass die erwartete Rechenzeit zur Lösung des Problems exponentiell mit der Größe der Eingabe wachsen kann. Obwohl Z3 fortschrittliche Heuristiken und Optimierungen verwendet, bleibt die grundsätzliche Herausforderung, dass die Lösung von NP-vollständigen Problemen in der Regel exponentielle Rechenzeit erfordert. Bei großen Instanzen kann dies dazu führen, dass die Rechenzeit für die praktische Problemlösung untragbar wird.

In der Praxis wird oft auf heuristische oder approximative Algorithmen zurückgegriffen, um große Instanzen von NP-vollständigen Problemen wie dem CVRP zu lösen. Diese Ansätze sind zwar meistens nicht die optimale Lösung, aber sie

liefern oft ausreichend gute Ergebnisse in einer praktikablen Zeit.

Es wurde eine beispielhafte Implementierung einer Lösung für das CVRP mithilfe des Z3-Solvers implementiert<sup>3</sup>. Die beispielhaften Daten mit der Problemgröße von 6 Haltestellen und 2 Fahrzeugen (Problemgröße 12) kann innerhalb kürzester Zeit (weniger als 1 Sekunde) gelöst werden. Ein CVRP mit der Problemgröße (Anzahl von Haltestellen multipliziert mit der Anzahl der Fahrzeuge) von 30 konnte in circa einer Sekunde gelöst werden. Die Problemgröße von 36 benötigt bereits eine Durchlaufzeit von 10 Sekunden. Bei einem beispielhaften Problem mit 16 Haltestellen und 3 Fahrzeugen, also einer Problemgröße von 48 beträgt die Durchlaufzeit bereits mehrere Minuten.

Daher konnte das kleinste Problem von Augerat (32 Haltestellen mit 5 Fahrzeugen, also einer Problemgröße von 160) nach einer Rechenzeit von mehreren Stunden nicht durch die Z3-Optimierungsroutinen gelöst werden. Daher würde das untersuchte Problem mit 80 Haltestellen und 10 Fahrzeugen (Problemgröße von 800) exponentiell länger dauern und kann nicht in einer realistischen Zeit gelöst werden.

### III. CONCLUSIO

Aufgrund der Eigenschaft von NP-vollständigen Problemen kann nicht sicher festgelegt werden, was die absolut beste Lösung ist, da die gewählten heuristischen Lösungsansätze zwar meist eine gute, jedoch niemals eine hundertprozentige beste Lösung für das gegebene Problem sicherstellen können. Für die gewählte CVRP Instanz liegt die beste bekannte Lösung bei einer Distanz von 1763m[5]. Diesem Ergebnis am nächsten kommt die Implementierung von Google durch die Google OR-Tools, welche eine Lösung mit einer Routenlänge von 1785m liefert, bei einer vordefinierten Durchlaufzeit von 120 Sekunden und den oben beschriebenen, von Google empfohlenen Parametern. Diese Lösung ist schon sehr nahe an der besten bekannten Lösung von 1763m. Der Offspring Selection Genetic Algorithm (OSGA) lieferte im Vergleich dazu ein bestes Ergebnis von 3339m und der Z3 Solver konnte das Problem aufgrund der Größe nicht lösen.

Durch den Z3 Solver konnte leider keine Lösung für das gewählte CVRP erreicht werden, da es sich beim CVRP um ein NP-Vollständiges Problem handelt und der Solver daher durch größere Problemgrößen exponentiell längere Durchlaufzeiten benötigt. Daher konnte die Funktionalität des Solvers selbst durch kleinere Probleme zwar bewiesen, das gewählte Problem mit der Größe von 80 Haltestellen und 10 Fahrzeugen jedoch nicht in einer zumutbaren Zeit gelöst werden.

### REFERENCES

- [1] *About OR-Tools*. 2023-01-09. URL: <https://developers.google.com/optimization/introduction>.
- [2] M. Affenzeller and S. Wagner. "Offspring Selection: A New Self-Adaptive Selection Scheme for Genetic Algorithms". In: *Adaptive and Natural Computing Algorithms*. Ed. by Bernardete Ribeiro et al. Vienna: Springer Vienna, 2005, pp. 218–221. ISBN: 978-3-211-27389-0.

<sup>3</sup><https://github.com/alx-hbl/FMT2-CVRP-eval/tree/main>

- [3] Philippe Augerat. “Approche polyédrale du problème de tournées de véhicules”. Theses. Institut National Polytechnique de Grenoble - INPG, June 1995. URL: <https://theses.hal.science/tel-00005026>.
- [4] *Capacity Constraints*. 2023-01-18. URL: <https://developers.google.com/optimization/routing/cvrp>.
- [5] HEAL. *Heuristic Lab (version 3.3.16)*. 2024. URL: <https://dev.heuristiclab.com/raw-attachment/wiki/Download/HeuristicLab%203.3.16.zip>.
- [6] Paolo Toth and Daniele Vigo. “1. An Overview of Vehicle Routing Problems”. In: *The Vehicle Routing Problem*, pp. 1–26. DOI: 10.1137/1.9780898718515.ch1. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9780898718515.ch1>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9780898718515.ch1>.
- [7] Stefan Winkler. *Computational Intelligence II*. 2024.