Konzeption eines Machine-Learnings-Verfahrens zum Lösen von Green Vehicle Routing Problemen

Pablo Stockhausen [1], Andreas Johannsen[2], Robert Maurer[3]

Abstract: Dieser Beitrag entwickelt ein Konzept zur praktischen Umsetzung eines Machine- Learning-Verfahrens zum Lösen von Vehicle Routing Problemen im Kontext einer nachhaltigen "Letzte-Meile"-Logistik, welches durch einen Prototyp umgesetzt und getestet wurde. Dabei wurden Aspekte von kombinatorischen Optimierungsalgorithmen in Form eines Ameisenalgorithmus zur Unterstützung des angewendeten Machine-Learning-Systems verwendet. Der Prototyp basiert auf einem "Reinforcement Learning"-System und verwendet als Algorithmus "REINFORCE mit Baseline". In einer Vergleichsanalyse wird der Prototyp mit dem bekannten Vertreter für kombinatorische Optimierungsalgorithmen, Google-OR, an Hand von zwei Anwendungsszenarien verglichen. Die kombinatorischen Optimierungsalgorithmen konnten sich hinsichtlich der Lösungsqualität gegenüber dem Prototyp durchsetzen. Dafür überzeugt der Prototyp in der Laufzeit und dem Automatismus. Es wurde festgestellt, dass der verwendete Machine-Learning-Kontext für kleine bis mittelgroße Logistikdomänen nur geringe Vorteile ergibt. Eine Verwendung von lernenden Systemen für das Vehicle Routing Problem empfiehlt sich bei einem größeren Stoppvolumen und einer erweiterten IT-Infrastruktur. Letztlich bietet der Prototyp in diesem Beitrag eine Alternative gegenüber dem Outsourcing an Drittanbietern zum Lösen von Routingproblemen.

Keywords: Vehicle Routing Problem; Machine Learning; Reinforcement Learning; Ant Colony Optimization; Letzte- Meile- Logistik; Ameisenalgortyhmus.

1 Universität Potsdam, Wirtschafts- und Sozialwissenschaftlichen Fakultät, August-Bebel-Straße 89, 14482 Potsdam, stockhausen017@gmail.com

2 Technische Hochschule Brandenburg, Fachbereich Wirtschaft, Magdeburger Str. 50, 14770 Brandenburg an der Havel, johannse@th-brandenburg.de>

3 Technische Hochschule Brandenburg, Fachbereich Wirtschaft, Magdeburger Str. 50, 14770 Brandenburg an der Havel, robert.maurer@th-brandenburg.de

https://doi.org/10.30844/AKWI_2022_17

1 Vehicle Routing Probleme im Kontext der „Letzten-Meile“-Logistik

1.1 Motivation und Zielsetzung

Im Jahr 2020 umfasste das Sendungsvolumen in Deutschland 4,05 Mrd. Paket-, Express- und Kuriersendungen mit einer in den kommenden Jahren steigenden Tendenz, wobei der Hauptanteil an Sendungen den B2C Bereich bedient hat [BU21]. Zu dem Anstieg an Paketsendungen und den damit auftretenden Routingproblemen, ist auch das wissenschaftliche Interesse für Vehicle Routing Probleme (VRP) auf Basis von modernen IT-Lösungen gestiegen. Eine Suche nach den Begriffen „Vehicle Routing“ in der Datenbank IEEE Explore (https://ieeexplore.ieee.org“) ergab über 6500 Treffer in den letzten zehn Jahren. Ungeachtet dessen, gibt es nur wenige wissenschaftliche Arbeiten im Bereich der Machine-Learning (ML) -Verfahren zum Lösen von VRP. Eine Suche mit Google Scholar (https://scholar.google.com) ergab 15 Veröffentlichungen in den letzten fünf Jahren, welche im Titel die Begriffe „Vehicle Routing“ und „Machine-Learning“ verwendet haben. Zusätzlich bewegen sich die fundamentierten Forschungserkenntnisse in einem überwiegenden theoretischen Umfeld mit Schwerpunkt auf mathematischen Modellen und Annahmen, jedoch mit wenig Praxisbezug. Die Hauptmotivation dieser Arbeit liegt darin, ein praktisches Konzept, in Form eines Prototyps, zum Lösen von VRP zu entwickeln und im Besonderen die Aspekte eines Capacitated Vehicle Routing Problems (CVRP) zu berücksichtigen. Dieser Beitrag verwendet CVRP und VRP gleichbedeutend. Weiterhin soll gezeigt werden, dass sich ML-Verfahren in einem produktiven Umfeld durchaus beweisen und eine langfristige Alternative zum Outsourcing an Drittanbieter darstellen können. Dies wird durch einen Vergleich, hinsichtlich der Effektivität unter ausgewählten Bewertungskriterien, zwischen ML-Verfahren und den klassischen Optimierungsalgorithmen zum Lösen von VRP dargestellt.

1.2 Thematische Hinführung

Das Vehicle Routing Problem (VRP) beschreibt im Gebiet der emissionsfreien „Letzte-Meile“-Großstadtlogistik ein kombinatorisches Optimierungsproblem, welches folgende Grundfrage behandelt: „Was ist der optimale Satz von Routen für einen bestimmten Fuhrpark, um eine bestimmte Anzahl von Kunden zu beliefern?“ [BRN15, S.1]. Das VRP wurde erstmalig von Dantzig & Ramser (1959) durch ihre wissenschaftliche Arbeit „The Truck Dispatching Problem“ diskutiert, wobei der Problemkontext die Lieferung von Kraftstoffen darstellte und mithilfe von algorithmischen Wegen gelöst wurde [DR59]. Zum Lösen von VRP eignen sich besonders Machine-Learning (ML)-Verfahren, da sie algorithmisch vorgehen und erfahrungsbasiert entscheiden können. Die Erfahrungen ergeben sich durch die in bestimmten Themendomänen auftretenden Abonnement-Strukturen und den daraus

resultierenden bekannten VRP-Instanzen. Der spezifische Anwendungsfall des VRP entsteht im Rahmen dieser Arbeit durch den Kontext einer „Letzten-Meile"-Großstadtlogistik. Dies ist eine moderne Form der urbanen Logistik und umfasst den letzten Logistik Schritt einer Paketzustellung. Im Besonderen wird ein emissionsfreier Ansatz angestrebt, der durch die Verwendung von Lastenrädern sowie Mikro-Depots realisiert wird und die Bezeichnung als Green-VRP begründet [RSC20, S. 6]. Der Aspekt einer „Letzten-Meile"-Großstadtlogistik spielt beim Lösen von kombinatorischen Optimierungsproblemen dahingehend eine Rolle, dass in der Regel von kurzen Routendistanzen ausgegangen werden kann und Touren durch die Verwendung von Lastenfahrrädern klaren Kapazitätsbeschränkungen unterliegen. Die Planung effizienter Touren ist somit nicht nur ein ausschlaggebender Faktor für das Ausmaß der hohen Logistikkosten einer emissionsfreien Lieferung, sondern auch nötig, um mit der Alternative von Diesel- bzw. Benzin-Kleintransportern mithalten zu können.

1.3 Herausforderungen & Wettbewerbsfähigkeit

Es bestehen Herausforderungen in der späteren Implementierung des Prototyps bezüglich des Wettbewerbs zwischen klassischen Optimierungsalgorithmen und des verwendeten ML-Verfahrens zum Lösen von VRP. Besonders im Bezugspunkt der Performance wird das gewählte ML-Verfahren des Prototyps anfänglich fordernder sein, da es mehr Prozessschritte, wie beispielsweise das Trainieren des Modells, durchlaufen muss. Der Prototyp ist dann wettbewerbsfähig, wenn: Er kurze Routen auf Basis der Tour länge bildet, die Ergebnisse in wenigen Sekunden bereitstellt, allgemein auf nicht bekannte VRP-Instanzen reagiert, auf allen VRP-Instanzen gleich gut arbeitet und kein manuelles Eingreifen benötigt wird.

2 Methodik

2.1 Auswahl des Machine-Learning -Verfahrens

Im Rahmen dieses Beitrages verwendet das ML-Modell ein Reinforcement Learning (RL)-System zum Lösen von VRP der „Letzte-Meile"-Großstadtlogistik [RA18]. Das RL unterscheidet sich maßgeblich von den Alternativen des Supervised- und Unsupervised-Learnings, da es eine andere Herangehensweise für den Aufbau eines Lernenden-Systems verwendet. Das lernende Modell beschreibt im RL einen Agenten oder auch Entscheidungsträger, der ein Environment beobachtet, Aktion darauf ausführt und deren Dynamiken, beginnend von einem unwissenden Startpunkt eigenständig lernt [RA18, S. 2]. Grundlegend steht $A_{(st)} \subseteq A$ für die Menge der Aktionen, die der Agent zu einem Zeitpunkt t ausführen kann [Lo20, S.15]. Dabei beschreibt $a_t \subseteq A$ die ausgewählte Aktion in einem Zeitpunkt t des

Agenten. Nach jeder Aktion kann der Agent zwei Arten von Belohnungen erhalten: Eine sofortige oder eine verzögerte Belohnung. Eine sofortige Belohnung findet Anwendung bei Aktionen des Agenten, die eine unmittelbare Beurteilung zu lassen. Diese wurde auch in dem entwickelten Prototyp verwendet. Zum Beispiel lässt sich das Überqueren einer roten Ampel unmittelbar bewerten, da ein negatives Verhalten direkt erkennbar ist und nicht erst durch Folgeverhalten erkenntlich wird. Unter einer verzögerten Belohnung kann beispielsweise eine Aktion in einer Schach- oder Go-Partie verstanden werden, weil die Belohnung sich an den Folgereaktionen misst. Diese wurde auch in der wissenschaftlichen Arbeit von Silver et al. (2017) verwendet [Si17]. Es wurde sich für RL-System entschieden, um ohne menschliches Vorwissen, Lösungen für komplexe Optimierungsprobleme zu finden und somit Entwicklungs- und Dispositionsaufwand einzusparen.

2.2 Trainings- und Ergebnisdaten

In Kooperation mit einem Unternehmen, welches sich auf nachhaltige Transportlösungen spezialisiert hat, konnten als Datengrundlage die anonymisierten Kundendaten der Ökobox-Anbieter in Berlin zum Trainieren und Testen des Modells verwendet werden. Die Datenobjekte verfügen über folgende Eigenschaften: Längengrad, Breitengrad, Stoppgewicht, Stoppvolumen und einem anonymisierten Identifikator. Die Datenobjekte sind für ein ausgewähltes Datum und einem bestimmten Micro-Depot spezifisch.

2.3 Prototyping

Zur Fundamentierung der Forschung und auf Basis der praktischen Motivation wurde ein qualitativ-konstruktivistischer Ansatz in Form des Prototypings ausgewählt. Der Prototyp soll in der Endfassung einen Service darstellen, der ML-Modelle trainiert, speichert und anwendet. Das ML-Modell ist gegenüber dem Versender, Micro-Depot, Lieferant, Wochentag, Fahrzeuggewicht, Fahrzeugvolumen sowie ML-Verfahren sensibel und wird auf Instanz Basis gespeichert. Das ML-Modell wird nur durch Stopps von einem Tag bis zu einer Woche trainiert, um zu vermeiden, dass Abonnement-Kunden durch doppeltes Auftreten die Strategie des Agenten verfälschen.

2.4 Vergleichsanalyse

Im Kontext dieses Beitrags wird eine Vergleichsanalyse zwischen dem entwickelten Prototyp und dem VRP-Solver von Google-OR durchgeführt. Der VRP-Solver basiert nach Google (2021) auf heuristischen Algorithmen, die als „First Solution Strategy“ kategorisiert werden und optional durch „Local-Search“-Strategien erweitert werden können [Go21]. Die Resultate aus dem Vergleich sollen den Prototypen evaluieren, Verbesserungsmöglich-

keiten aufdecken und das Potenzial einer Implementierung von Reinforcement Learning (RL)-Verfahren zum Lösen von VRP in Erwägung ziehen. Die Analyse ist in zwei Teile gegliedert. Im ersten Teil der Vergleichsanalyse liegt der Hauptfokus auf einer Effektivitätseinschätzung des entwickelten Prototyps mit der zugrundeliegenden Basisfunktionalität des ausgewählten RL-Algorithmus „REINFORCE mit Baseline“ zum Lösen von VRP. Hiernach beruht die Effektivitätseinschätzung auf den Schwerpunkten der Trainings- sowie Testlaufzeit, aber auch der Vergleichskriterien der Distanz und des zeitlichen Aufwandes. Der Distanzaufwand wird mit der Haversine-Formel belegt. Hierbei wird der Zeitaufwand durch die hinterlegten Fahrzeuggeschwindigkeiten, die Distanz und einer definierten Stoppverweildauer von fünf Minuten errechnet. Weitere Vergleichskriterien für ein aufbauendes Experiment wären die Profitabilität, Service-Qualität, Konsistenz sowie externe Faktoren [VLM19, S. 2]. Es wird ein Stoppvolumen von 430 Stopps verteilt auf vier Wochen im Mai 2021 betrachtet, wobei das ML-Modell nach jedem Tag mit 200 Iterationen trainiert wird. Es wird aufbauend trainiert, sodass jede folge Woche auf die Erfahrung der Vorangegangenen aufbaut. Im zweiten Teil der Analyse liegt das Augenmerk auf dem Verbesserungspotenzial des ML-Verfahrens und dem Ausarbeiten von Anpassungsimpulsen, die aus dem vorherigen ersten Teil abgeleitet werden. Das Ziel ist es, sich so konvergent wie möglich an den errechneten Distanz- sowie Zeitaufwand der Konkurrenz anzunähern, um dadurch mit einer besseren Laufzeit zu überzeugen. Im zweiten Teil wird derselbe Problemkontext für einen aussagekräftigeren Vergleich betrachtet. Hierbei wird analog zum ersten Teil nach jedem Tag mit angepassten 115 Iterationen trainiert. Die Iterationsanzahl wurde auf 115 herabgesetzt, um mögliches „overfitting“ des ML-Modells gegenüber den Trainingsdaten zu verhindern.

3 Ergebnisse der Forschungsarbeit

3.1 Modellierung und Konzeption des Machine-Learning-Modells

Das ML-Modell beruht grundlegend auf einem Markov Decision Process (MDP) in Verbindung mit einem Ant Colony Optimization (ACO)-Algorithmus. Dies ist ein Verfahren der kombinatorischen Optimierung, wobei die Grundsteine des Modells durch Hyeong Soo Chang et al. (2004) gelegt wurden [Ch04]. Das ACO-Verfahren wurde im Trainingsmodus des Prototyps als erste Prozessinstanz eingesetzt, um dem RL-Verfahren im anschließenden Prozessschritt eine Vorahnung der Übergangswahrscheinlichkeitsverteilung bereitzustellen und schon trainierte ML-Modelle für den betrachteten Problemkontext einzustimmen. Vergleichsweise wurde ein ähnliches Unterfahren von Yuan Sun et al. (2020) durchgeführt, welche aber als Ausgangslage ein durch ML unterstützendes ACO-Modell betrachtet haben [Su20]. Das ML-Modell wird durch ein MDP definiert. Bei Markov-Modellen sind der Folgezustand und die RL-Belohnung nur vom momentanen Zustand und der gewählten Aktion des Agenten abhängig [Lo20, S.15]. Das angepasste MDP in dem entwickelten Prototyp wurde in Anlehnung an Puterman (2014) modelliert [Pu14]. Als ML-Algorithmus

verwendet der Prototyp den „REINFORCE“-Algorithmus oder auch Monte-Carlo Policy-Gradient, um eine optimale Policy π* zu finden. Für die Entwicklung des Prototyps und der Anwendung des „REINFORCE“-Algorithmus wurde sich an dem Buch „Reinforcement Learning – An Introduction“ von Sutton & Barto (2018) orientiert [SB18]. Dabei zeichnen sich Monte-Carlo-Methoden dadurch aus, dass kein ganzheitliches Wissen über das Environment für die Findung einer optimalen Policy nötig ist. Dies liegt daran, dass das System von den Interaktionen mit dem Environment lernt [SB18, S. 91]. Der „REINFORCE“-Algorithmus des Prototyps baut auf einer parametrisierten Policy $\pi (a \mid s, \theta)$ auf und verwendet zusätzlich einen Vergleich zwischen den momentanen Aktionswerten und einer willkürlichen Baseline. In abgewandelter Form ist der „REINFORCE mit Baseline“-Algorithmus nach Sutton & Barto (2018) konstruiert und wurde in Abb. 1 dargestellt [SB18, S. 330]. Eine Baseline kann dabei eine zufällige Variable sein oder wie im Rahmen dieses Prototyps eine „state-value“-Funktion, die eine Bewertung über die möglichen Belohnungen des gesamtem Zustandsraumes widerspiegelt [SB18, S. 329].

Data : Iterationen k, parametrisierte Policy $\pi(a|s,\theta)$, parametrisierte „state-value“-Funktion (baseline) $V(s,w)$
Result : optimal parametrisierte Policy π^*
initialisieren der Policy-Parameter θ;
initialisieren der Gewichte w;
for $i \leftarrow 0$ **to** k **by** 1 **do**
  $Eps \leftarrow$ Episode nach $\pi(\cdot|\cdot,\theta) = \{S_0, A_0, R_1, ..., S_{T-1}, A_{T-1}, R_T\}$;
  foreach *step* $t \in Eps,\ t = 0, 1, ..., T-1$ **do**
    $G \leftarrow \sum_{j=t+1}^{T} \gamma^{j-t-1} \times R_j$;
    $\delta \leftarrow G - V(S_t, w)$;
    $w \leftarrow w + \alpha \times \gamma^t \times \delta \times \nabla V(S_t, w)$;
    $\theta \leftarrow \theta + \alpha \times \gamma^t \times \delta \times \nabla ln\pi(A_t|S_t, \theta)$;
  end
end

Abb. 1 „REINFORCE mit Baseline“-Algorithmus in Anlehnung an Sutton und Barto (2018) [SB18, S.330]

3.2 Prototypenentwicklung

Das Backend des Prototyps wird in der Programmiersprache Python entwickelt und das Web-Frontend, welches zum besseren Debugging nötig ist, in React. Es wurden bewusst keine vorhandenen ML-Frameworks wie beispielsweise Keras, Tensorflow oder PyTorch verwendet. Die Entscheidung liegt darin begründet, dass im Falle der Verwendung von

komplexen ML-Frameworks das Risiko einer Abhängigkeit von dem jeweiligen Framework-Support bestünde. Darüber hinaus zeigt sich ein besserer Verständnis- und Erfahrungsgewinn bei einer Eigenproduktion von komplexen ML-Verfahren gegenüber dem Einsetzen verschiedener ML-Frameworks, wobei eher dazu tendiert wird, das Framework zu lernen und nicht das darauf aufbauende ML-Verfahren. Weiterhin traten bei der Entwicklung des Prototyps spezielle Herausforderungen auf. Anführend dafür ist die Thematik des lokalen Minimums, in dieser sich der Prototyp in den Anfangsphasen der Entwicklung häufig verfangen hat. In Abb. 2 ist ein Beispieldurchlauf der kumulierten Belohnungen, welche als Gesamtlänge in km aller gebildeten Touren in einer Episode verstanden werden, über 2000 Episoden dargestellt. In diesem Fall betrachtet der Agent ein Problemkontext von 19 Stopps verteilt in Berlin. Dabei findet er zum Ende hin eine gute Lösung, wobei ihm sich die Beste Lösung nicht erschließt, obwohl er diese bereits in den ersten 250 Iterationen entdeckt hatte. Dies lag zum einen daran, dass der Lernfaktor zu niedrig eingestellt war, aber zum anderen der Erkundungsfaktor den Agenten nicht genug zum erweiterten Erkunden des Environments beeinflusst hatte. Zur besseren Erkundung wurde sich einer angepassten dynamischen Epsilon-Erkundung bedient [SB18, S.30].

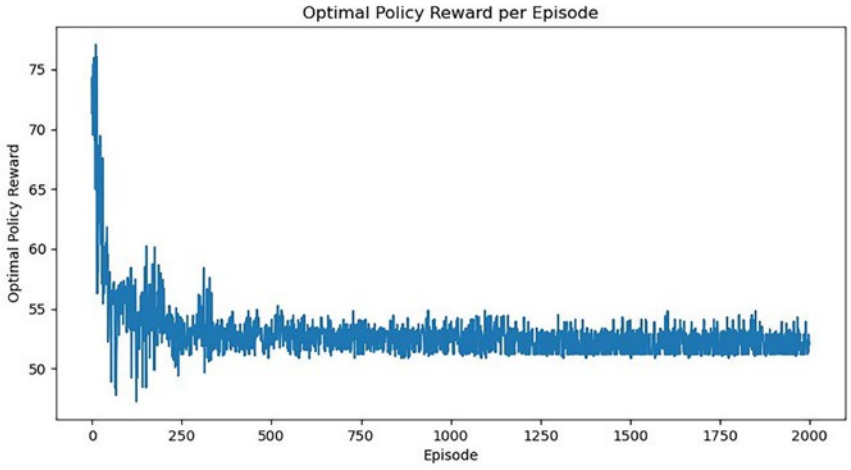

Abb. 2 Beispiel Durchlauf eines ML-Verfahren im lokalen Minimum

4 Interpretation

4.1 Evaluation der Ergebnisse

Im ersten Teil der Vergleichsanalyse und der verwendeten Basis Implementation ist zu erkennen, dass das ML-Verfahren in den ausgewählten Bewertungspunkten schlechter abgeschlossen hat als der Konkurrent. Ein Durchsetzen anhand der Laufzeit ist zwar möglich, dafür sollten aber die Touren nicht deutlich mehr Distanzaufwand in Anspruch nehmen. Die Bezahlung im Bereich der „Letzten-Meile"-Logistik erfolgt in der Regel pro ausgefahrenen Stopp. Aufgrund dessen wollen die Fahrer nicht durch schlecht optimierte Routen gebremst werden. Der Prototyp mit der Grundlagen Implementation von „REINFORCE mit Baseline" erkennt im ersten Analyseschritt teilweise nahe gelegene Stopps nicht und wählt dadurch überflüssig längere Strecken aus. Er weicht im Durchschnitt +6,94 km und +14,13 min von der Lösung des Konkurrenten ab. Die Resultate des zweiten Analyseschritts zeigen eine Steigerung in den Bewertungspunkten gegenüber der ersten Version. Die Abweichungen der kumulierten Distanzen für die gebildeten Touren an den jeweiligen Tagen sind in Abb. 3 dargestellt. Die durchschnittlichen Abweichungen zwischen der Prototyp Lösung und OR Strategie liegt bei +6,51 km +11,01 min.

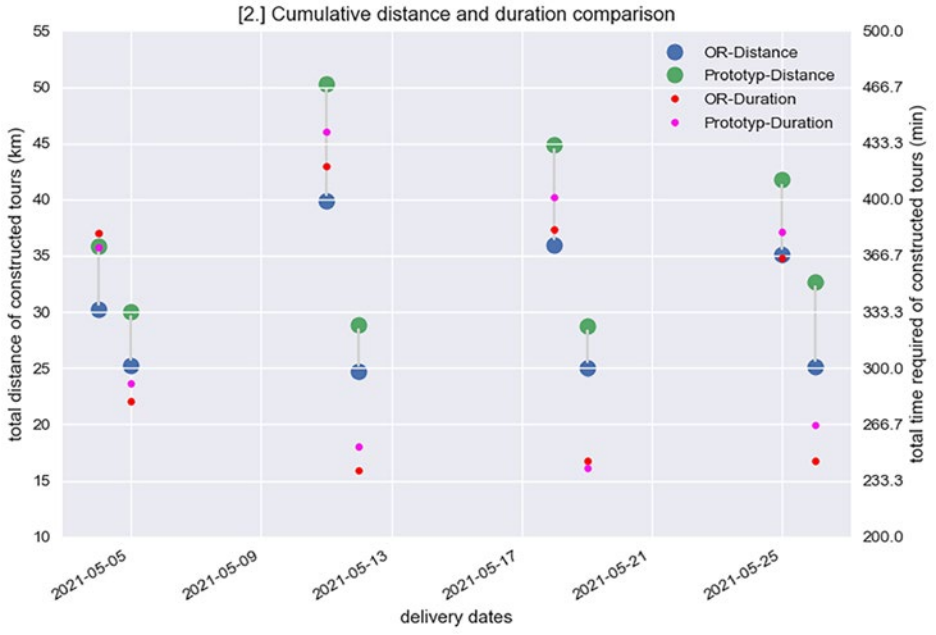

Abb. 3 Vergleich des kumulierten Distanz- und Zeitaufwandes zwischen OR Lösung und erweiterter Prototyp Lösung im zweiten Analyseschritt

Der Prototyp ist auch im zweiten Teil der Analyse bezüglich der Laufzeit zum Lösen von VRP-Instanzen deutlich schneller als der Konkurrent, weil er erfahrungsbasierte Entscheidungen treffen kann. Der nötige zeitliche Trainingsaufwand für den zweiten Analyseschritt wurde in Abb. 4 dargestellt.

Abb. 4 Vergleich des Trainings-Laufzeitverhaltens zwischen OR und der Prototyp Lösung im zweiten Analyseschritt

Dieser ist für eine produktive Umgebung in Relation zu der Stoppanzahl noch zu aufwendig und sollte im Hinblick auf die Wettbewerbsfähigkeit optimiert werden. In Hinsicht auf die genannten Herausforderungen und damit indirekten Anforderungen an die Wettbewerbsfähigkeit des Prototyps, im Gliederungspunkt 1.3, lassen sich folgende resultierende Punkte, welche der Prototyp in der Lage ist zu erfüllen bzw. nicht zu erfüllen, ableiten:

- Nach Durchlauf des Trainingsprozesses ist der Prototyp in der Lage eine gute bis sehr gute Lösung zu finden.
- Der Prototyp kann ohne manuelles Eingreifen Lösungen erarbeiten.
- Im Testmodus wird das Ergebnis des Prototyps in Millisekunden bereitgestellt.
- Der Prototyp schneidet optimal auf hauptsächlich bekannten VRP-Instanzen ab. Sofern der überwiegende Teil der VRP-Instanz dem Prototyp nicht bekannt ist, wird der Prototyp keine annähernd optimale Lösung finden.

- Aufgrund dessen, dass der Prototyp auf nicht bekannte VRP-Instanzen schlechter reagiert, ist es auch nicht gegeben, dass er auf allen VRP-Instanzen grundlegend gleich arbeitet.

4.2 Vergleich zu kombinatorischen Optimierungsalgorithmen

Die Vergleichsanalyse ergab, dass das ML-Verfahren in seiner jetzigen Implementation nicht das gleiche Niveau der Lösungsqualität für VRP-Instanzen aufweisen kann wie Optimierungsalgorithmen. Des Weiteren zeigten die ausgewählten Anpassungsimpulse für die Basis Implementation von „REINFORCE mit Baseline“ keine signifikante Verbesserung. Die Anpassungsimpulse erreichten das der Prototyp in weniger Iterationen eine bessere Lösung erzielte. Die Verbesserung der Lösung weicht aber nur geringfügig von der Basis Implementation ab. Ungeachtet dessen weisen Vergleichsstudien von RL-Verfahren mit verschiedenen Bewertungskriterien und gleicher Laufzeit selten eine optimale Lösung gegenüber kombinatorischer Optimierungsalgorithmen auf [Ma20]. Dies wird damit begründet, dass es das grundlegende Ziel von RL-Verfahren ist, sowohl schlechte Lösungen zu vermeiden als auch eine durchschnittlich gute Lösung zu erzielen. Folglich wird somit auch das Ziel der Prototypen definiert. Im Gegensatz zu dem Prototyp betrachtet der Google-OR-Solver die Struktur des Problems ganzheitlich und erreicht, mit genug Laufzeit und Rechenleistung, asymptotisch eine optimale Lösung. Neben Google-OR gibt es auch andere Alternativen für das Lösen von kombinatorischen Optimierungsproblemen wie beispielsweise Concorde TSP Solver (https://www.math.uwaterloo.ca/tsp/concorde.html) oder die Dienste von openrouteservice (https://openrouteservice.org/).

4.3 Relevanz für die „Letzte-Meile“-Logistik

Im Bezugspunkt der Relevanz des Prototyps für die „Letzte-Meile“-Logistik und unter der Rücksprache kooperierenden Unternehmen, lassen sich folgende Resultate ableiten. Der Prototyp ist für ein geringes Stoppvolumen von weniger als 1200 Stopps pro Tag nicht effizient genug. Eine der Begründungen ist, dass der manuelle Dispositionsaufwand und das Einsetzen von kombinatorischen Optimierungsalgorithmen für weniger als 400 Stopps in einem kleinem „Letzte-Meile“ Gebiet deutlicher geringer sind als der Aufwand des Trainierens des Prototyps. Erst ab 400 Stopps schwächelt der Google-OR-Solver in der Laufzeit mit 10 s. Unter der Retrospektive und dem Feedback des kooperierenden Unternehmens, wurde die Entscheidung des RL für die „Letzte-Meile“ überdacht und Ideen bezüglich anderer ML-Systeme ausgearbeitet. Ein Favorit ist es Supervised-Learning einzusetzen, um die Dispositionsart der Zulieferer-Disponenten zu lernen, sodass die ML-Komponente unterstützend wirkt und nicht automatisiert VRP-Instanzen löst.

5 Zusammenfassung und Ausblick

Der vorliegende Beitrag bietet ein Konzept zur Basis Implementation eines Reinforcement Learning (RL)-Verfahrens, in Form von „REINFORCE mit Baseline“ kombiniert mit einem Ant Colony Optimization Algorithm (ACO) zum Lösen eines Vehicle Routing Problems (VRP). Darüber hinaus stellt der entwickelte Prototyp auf Basis der Erweiterungsimplementierung von „REINFORCE mit Baseline“ eine Alternative gegenüber Drittanbietern wie beispielsweise Google-OR da. Des Weiteren werden Impulse zum adaptiven Lösen von VRP mittels verschiedener Optimierungsmechanismen innerhalb des Prototyps geboten. Mit Ausblick auf die weitere Forschungslandschaft von Machine-Learning (ML) und VRP haben die Untersuchungsergebnisse gezeigt, dass eine erweiterte Implementation von RL-Verfahren ein gutes bis sehr gutes Ergebnis im Kontext von einem NP-schweren Problem hinsichtlich der untersuchten Bewertungspunkte und gegenüber den klassischen Optimierungsverfahren erzielen können. Für die weitere Zukunft des Prototyps wäre ein Reduzieren der Komplexität der einstellbaren Parameter denkbar, um mögliches „overfitting“ zu vermeiden. Eine Erweiterbarkeit des Prototyps hinsichtlich anderer RL-Verfahren wäre möglich und bedeutsam, um die Resultate zu validieren. Eine Erkundung weiterer RL-Verfahren ist dahingehend wichtig, da im Gebiet von RL die geringsten Änderungen der Parameter oder des verwendeten Verfahrens starke Abweichungen im resultierenden Ergebnis auslösen. Es wäre in diesem Zusammenhang lohnend zu untersuchen, wie sich das Implementieren von Proximal Policy Optimization (PPO), welches eine neue Art von Policy-Gradienten-Methoden im RL beschreibt, auf die Lösungsqualität auswirkt. Dabei zeichnet sich PPO vor allem durch eine einfachere Implementation und einen einheitlichen Aufbau aus [Sc17].

Literaturverzeichnis

[BRN15] Braekers, K., Ramaekers, K. & Nieuwenhuyse, I.: „The Vehicle Routing Problem: State of the Art Classification and Review“, Computers & Industrial Engineering, Vol. 99, 2015.

[BU21] Bundesverband Paket und Expresslogistik e. V. (BIEK) (Hg.) (2021) KEP-Studie 2021 – Analyse des Marktes in Deutschland: Eine Untersuchung im Auftrag des Bundesverbandes Paket und Expresslogistik e. V. (BIEK), KE-CONSULT Kurte&Esser GbR [Online]. Verfügbar unter https://www.biek.de/download.html?getfile=2897 (Stand: 17.08.2021).

[Ch04] Chang, H. S., W. J. Gutjahr, Yang, J. & Park S.: „An ant system approach to Markov decision processes“, Proceedings of the 2004 American Control Conference, 3820-3825 vol.4, 2004.

[DR56] Dantzig, G. B. & Ramser, J. H.: „The Truck Dispatching Problem“, Management Science, Vol. 6, No. 1, S. 80–91 [Online]. Verfügbar unter http://www.jstor.org/stable/2627477, 1959.

[Go21] Google (2021) Routing Options [Online]. Verfügbar unter https://developers.google.com/optimization/routing/routing_options#local_search_options (Stand: 01.06.2021).

[Lo20] Lorenz, U.: Reinforcement Learning, Springer Berlin Heidelberg, 2020.

[Ma20] Mazyavkina, N., Sviridov, S., Ivanov S. & Burnaev E.: „Reinforcement Learning for Combinatorial Optimization: A Survey“, CoRR, abs/2003.03600, 2020.

[Pu14] Puterman, M. L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming [Online], Wiley. Verfügbar unter https://books.google.de/books?id=VvBjBAAAQBAJ, 2014.

[RA18] Richard, S. S. & Andrew, G. B.: Reinforcement Learning: An Introduction, A Bradford Book, 2018.

[RSC20] Richter, R., Söding, M. & Christmann, G.: Logistik und Mobilität in der Stadt von morgen. Eine Expert*innenstudie über letzte Meile, Sharing-Konzepte und urbane Produktion, 2020.

[SB18] Sutton, R. S. & Barto, A. G.: Reinforcement Learning: An Introduction [Online], The MIT Press. Verfügbar unter http://incompleteideas.net/book/the-book-2nd.html, 2018.

[Sc17] Schulman, J., Wolski, P., Dhariwal, P., Radford, A., & Klimov, O.: „Proximal Policy Optimization Algorithms“, CoRR, abs/1707.06347, 2017.

[Si17] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T. & Hassabis, D.: „Mastering the game of Go without human knowledge“, Nature, Vol. 550, 354, 2017.

[Su20] Sun, Y., Wang S., Shen, Y., Li, X., Ernst, A. T., & Kirley, M.; Boosting Ant Colony Optimization via Solution Prediction and Machine Learning [Online], 2020.

[VLM19] Vidal, T., Laporte, G. & Matl, P.: „A concise guide to existing and emerging vehicle routing problem variants“, European Journal of Operational Research, Vol. 286, 2019.