

Bergische Universität Wuppertal
Fakultät für Wirtschaftswissenschaft
Schumpeter School of Business and Economics
Dissertation

Komplexitätstheoretische Analyse des Line-TSPs
mit kombiniertem Drohneneinsatz

Eingereicht von:

M.Sc. Anna Katharina JANISZCZAK

Gutachter:

Prof. Dr. Stefan BOCK

Lehrstuhl für Wirtschaftsinformatik und Operations Research

Prof. Dr. Dirk BRISKORN

Lehrstuhl für Produktion und Logistik



Danksagung

Diese Arbeit ist während meiner Beschäftigung als wissenschaftliche Mitarbeiterin am Lehrstuhl Wirtschaftsinformatik und Operations Research von Prof. Dr. Stefan Bock an der Fakultät für Wirtschaftswissenschaft der Bergischen Universität Wuppertal entstanden.

Ich möchte die Gelegenheit nutzen, um all jenen zu danken, die mich bei der Erstellung dieser Arbeit unterstützt und motiviert haben.

Zuallererst gilt mein aufrichtiger Dank meinem Betreuer, Prof. Dr. Stefan Bock, für die freundliche Unterstützung und die vielfältigen Anregungen, die mir den Zugang zu dieser Thematik eröffnet haben. Die zahlreichen Diskussionen auf fachlicher und persönlicher Ebene habe ich stets als bereichernden und konstruktiven Austausch geschätzt.

Mein Dank gilt auch Prof. Dr. Dirk Briskorn für seine Bereitschaft, als Zweitgutachter mitzuwirken.

Das gesamte Lehrstuhlteam sowie ehemalige Mitarbeiter des Lehrstuhls haben mich stets motiviert und hatten immer ein offenes Ohr für mich. Vielen Dank für die konstruktiven Gespräche.

Die Fertigstellung dieser Arbeit wurde auch durch die vielfältige Unterstützung meiner Familie ermöglicht. Mein besonderer Dank gilt meinen Eltern und meinem Partner für ihre Geduld und ihre liebevollen und motivierenden Worte.

Inhaltsverzeichnis

1	Einleitung	1
2	Literaturüberblick	4
3	Modelle	6
3.1	Was ist eine Drohne?	6
3.2	Gemischt-ganzzahliges lineares Programm	6
3.2.1	Parameter	6
3.2.2	Variablen	7
3.2.3	Modell	9
3.2.4	Anzahl aller partiell zulässigen Lösungen	15
3.3	Constraint Programming Ansatz	19
4	Komplexität verschiedener Varianten	23
4.1	Grundlegende Begriffe der Komplexitätstheorie	23
4.2	Keine Zeitfenster und $v^D \leq v^T$	26
4.3	Zeitfenster und $v^D = 0$ oder $v^T = 0$	35
4.4	Keine Zeitfenster und $v^D > v^T$	41
4.5	Komplexität mit Deadlines und $v^D > v^T$	79
4.6	Komplexität mit Release Dates und $v^D > v^T$	83
4.7	Komplexität mit Deadlines und $v^T > v^D$	89
4.8	Komplexität mit Release Dates und $v^T > v^D$	99
4.9	Komplexität des diskreten Falls	106
5	Abschluss und Ausblick	130
	Literatur	131

Abbildungsverzeichnis

3.1	Line-TSPwD Instanz	7
3.2	Lösung für gegebene Reihenfolgefestlegung a	15
3.3	Anzahl aller Möglichkeiten	16
3.4	Constraint Programming Lösung	22
4.1	Drohne bedient Kunde 1 und Kunde n	27
4.2	Höchstens ein Drohnenkunde bei $v^D \leq v^T$	28
4.3	Ort-Zeit Diagramm der optimalen Lösungen der beiden Instanzen	31
4.4	Ort-Zeit Diagramm der optimalen Lösungen der beiden Instanzen	34
4.5	Begegnungs- und Trennungszeitpunkte	42
4.6	Identischer Begegnungszeitpunkt LKW Hin- und Rückrichtung . .	53
4.7	Intervalle der Trennungszeitpunkte	57
4.8	Differenz Trennungszeitpunkt und Begegnungszeitpunkt	59
4.9	Hinflug Wendekunde nach t_{turn}	60
4.10	Lösung der Line-TSPwD-Instanz $\Psi(I)$	71
4.11	Strenge NP-Schwere bei Vorliegen von Deadlines	80
4.12	Teil einer Lösung nach der Reduktion	87
4.13	Line-TSPwD Instanz Reduktion mit $m = 4$ und $B = 7$	92
4.14	Eine 3-elementige Menge einer 3-Partitionsinstanz mit $B = 23$. .	96
4.15	Line-TSPwD Instanz Reduktion mit $m=5$ und $B = 6$	103
4.16	Erzwungene LKW-Tour	111
4.17	Zwei Drohnen-Operationen ab p_k^{en}	112
4.18	Lösung einer Instanz nach Reduktion	118
4.19	Schnitt zweier Kreise im \mathbb{R}^n bzgl. der p -Norm	120
4.20	Schnittmenge dreier Kreise in \mathbb{R}^2 bezüglich der 1-Norm	121
4.21	Kunden der DRS Instanz	122
4.22	Positionen der LKW-Kunden	125
4.23	LKW-Tour und Drohnen-Tour für ein Zeitsegment	129

Kapitel 1

Einleitung

Zu den bekanntesten Problemen der kombinatorischen Optimierung zählt das Traveling-Salesman-Problem (TSP), welches eine Tour in einem Graphen $G = (V, E)$ mit Kantengewichten c_e für alle $e \in E$ sucht, sodass alle Knoten genau einmal besucht werden und die Summe der Kantengewichte der Tour minimal ist. Sofern keine weiteren Bedingungen an den Graphen geknüpft sind, ist das Problem \mathcal{NP} -schwer im strengen Sinne (Karp, 1972), (Garey & Johnson, 1979). Für Graphen mit speziellen Strukturen besteht jedoch die Möglichkeit, dass ein solches Problem einfacher zu lösen ist. Dies bedeutet, dass es dazu einen Algorithmus mit asymptotisch polynomieller Laufzeit gibt. Ein Beispiel ist das sogenannte Line-TSP with Time Windows (Line-TSPTW) (Tsitsiklis, 1992), bei dem Kunden (Knoten) $1, \dots, n$ gegeben sind und nur ungerichtete Verbindungskanten zwischen den Kunden i und $i + 1$ für alle $i \in \{1, \dots, n - 1\}$ existieren. Dies bedeutet, dass der Graph als eine Linie aufgefasst werden kann. Bei einem ausgezeichneten Kunden $i^* \in \{1, \dots, n\}$ befindet sich zum Startzeitpunkt 0 ein LKW, der zu diesen Kunden fährt, um sie zu bedienen. In der Literatur wird der LKW auch als Server bezeichnet. Jeder Kunde hat ein Zeitfenster, also einen frühesten Zeitpunkt (Release Date), zu dem ein Kunde bedient werden kann, sowie einen spätesten Zeitpunkt (Deadline), zu dem ein Kunde bedient sein muss. Das Ziel ist, möglichst früh alle Kunden bedient zu haben. Eine Rückkehr zum Startkunden wird nicht vorausgesetzt, d. h. es wird eine offene Tour gesucht. Anwendung findet das Line-TSPTW zum Beispiel an Küstenlinien, Flüssen oder entlang einer Straße. Werden keine Zeitfenster vorausgesetzt, d. h. alle Kunden haben jeweils ein Release Date von 0 und eine Deadline von ∞ , so ist das Problem einfach zu lösen: Der LKW bewegt sich vom Startkunden zu Kunde 1, falls dieser näher an Kunde i^* liegt, andernfalls zu Kunde n und anschließend zum anderen Ende der Linie. Die Einbeziehung beliebiger Zeitfenster hat großen Einfluss auf die Komplexität des Line-TSPTW. In diesem Fall ist das Line-TSPTW \mathcal{NP} -schwer im strengen Sinne ((Tsitsiklis, 1992)). Sofern ausschließlich Deadlines oder ausschließlich Release Dates einzuhalten sind, lässt sich das Problem wiederum polynomiell lösen ((Psaraftis, Solomon, Magnanti & Kim, 1990), (Tsitsiklis, 1992)). Wenn zu den Deadlines noch Bedienzeiten (Processing Times) existieren, d. h. wenn sich das Fahrzeug für jeden Kunden eine zusätzliche Zeit bei diesem Kunden befindet, ist das Line-TSPTW \mathcal{NP} -schwer (Bock & Klamroth, 2013).

Im Rahmen der vorliegenden Arbeit erfolgt die Erweiterung des Line-TSPTW

um eine Drohne, die ebenfalls in der Lage ist, Kunden zu bedienen. Jedoch muss die Drohne, nachdem sie einen Kunden bedient hat, zum LKW zurückkehren, um einen nachfolgenden Kunden bedienen zu können. Jedem Kunden i auf der Linie ist eine Position $p_i \in \mathbb{R}$ zugewiesen. Dabei sind die Distanzen zwischen zwei Kunden der Absolutbetrag der Differenzen der Positionen. Dieses Problem wird als Traveling Salesman Problem on the Line with Drone transports (Line-TSPwD) bezeichnet. Die Komplexität verschiedener Szenarien wird hier analysiert. Dabei werden die Fälle unterschieden, ob entweder die Drohne mit Geschwindigkeit v^D oder der LKW mit Geschwindigkeit v^T schneller ist, Zeitfenster oder keine Zeitfenster vorhanden sind, oder der Start von Drohne und LKW ganz links oder ganz rechts von der Linie aus erfolgt oder innerhalb der Linie liegt. Darüber hinaus wird zwischen dem kontinuierlichen Fall und dem diskreten Fall unterschieden. Im kontinuierlichen Fall kann die Drohne an jedem Ort entlang der Linie aufgelesen und gestartet werden. In dem anderen Fall ist das Starten und Auflesen der Drohne lediglich bei Kundenorten möglich. Die einfache Struktur des Line-TSPwD erlaubt es, die in der vorliegenden Arbeit bewiesenen Ergebnisse auf andere Probleme zu übertragen, um deren Komplexität zu verifizieren. Ein Beispiel hierfür ist das sogenannte Drone Routing Subproblem (siehe (Tang, Hovee & Shaw, 2019)), das zwar schon im Allgemeinen \mathcal{NP} -schwer im strengen Sinne ist, jedoch wird für die Reduktion dort eine sehr spezielle Struktur der Abstände der Kunden genutzt. Aus den Ergebnissen der vorliegenden Arbeit lässt sich folgern, dass das Problem selbst dann \mathcal{NP} -schwer im strengen Sinne ist, wenn die LKW-Tour auf einer Linie liegt und die zu bedienenden Kunden sich ebenfalls auf der gleichen Linie befinden. Hierbei kann eine beliebige p -Norm als Abstandsmaß verwendet werden.

Bei der Auslieferung in einem städtischen Gebiet liegt häufig auf einer geringen Anzahl von Straßen jeweils eine größere Anzahl von Kunden, die bedient werden sollen. Dabei ist es möglich, dass die jeweiligen Straßen nur wenige Zufahrtmöglichkeiten haben (Bock, 2024). Das Line-TSPTW wäre für diese Struktur ein Spezialfall. Analoges gilt für die Erweiterung solcher Probleme um eine Drohne, welche die Auslieferung unterstützt.

Die Erweiterung von TSPs und Vehicle Routing Problemen (VRPs) um neue Zustelltechnologien wie Drohnen oder Lieferroboter gewinnt zunehmend an Bedeutung, da diese autonomen Fahrzeuge das Lösen des Problems der letzten Meile (last-mile delivery) unterstützen können. Bei diesem Problem handelt es sich um den letzten Schritt in der Lieferkette, also um den letzten Abschnitt des Transports von Waren bis zur Haustür des Kunden, der bei einem hohen Straßenverkehrsaufkommen schwieriger und zeitintensiver zu bewältigen ist. Die Gesamtmenge urbaner Güterverkehrsleistungen betrug im Jahr 2010 weltweit ca. 10 Billionen tkm (Tonnenkilometer), für das Jahr 2050 wird ein Aufkommen von ca. 30 Billionen tkm prognostiziert (Little, 2018).

Auslieferungsdrohnen werden bereits in verschiedenen Bereichen eingesetzt. So hat z.B. in Ruanda das Startup-Unternehmen Zipline zwei Flugplätze für Drohnen errichtet, die medizinische Produkte transportieren (Kaufmann, 2019). Von diesen Flugplätzen aus können die Drohnen jeden Teil des Landes erreichen. Diese Art des Arzneimitteltransportes wird mittlerweile auch in Deutschland getestet (Pharmazeutische Zeitung, 2021). Die deutsche Bundesregierung zeigt sich ebenfalls allgemein interessiert an dem Einsatz von Drohnen und hat diesbezüglich einen Aktionsplan aufgestellt. Hierbei sollen Forschung und der Übergang in die Praxis gefördert werden (BMVI, 2020).

Neben der Logistik im Gesundheitswesen gibt es auch Versandhändler, welche die Lieferung per Drohne planen (Amazon, 2018). Das Logistikunternehmen UPS testete 2017 die Auslieferung von Paketen mithilfe von Drohnen, die vom Dach des Auslieferungs-LKWs gestartet werden und nach der Zustellung zurück zum LKW fliegen, während der LKW weiterhin Pakete verteilt (Burns, 2017). Obwohl Drohnen erst seit kurzer Zeit tatsächlich im Logistikbereich im Einsatz sind, gibt es zahlreiche Forschungsartikel, um ihren Einsatz zu optimieren. Darauf wird in Kapitel 2 näher eingegangen.

In Kapitel 3 werden zwei Modelle vorgestellt, die das Line-TSPwD mathema-

Geschw.	kont./disk.	Deadl.	Release D.	$i^* \in \{1, n\}$	Komplexität
$v^T \geq v^D$	kont./disk.	Nein	Nein	Nein	polynomiell
$v^T \geq v^D$	kont./disk.	Ja	Nein	Nein	str. \mathcal{NP} -schwer
$v^T \geq v^D$	kont./disk.	Ja	Nein	Ja	polynomiell
$v^T \geq v^D$	kont./disk.	Nein	Ja	Ja	str. \mathcal{NP} -schwer
$v^T < v^D$	kont.	Nein	Nein	Ja	polynomiell
$v^T < v^D$	kont.	Nein	Nein	Nein	\mathcal{NP} -schwer
$v^T < v^D$	kont.	Ja	Nein	Ja	str. \mathcal{NP} -schwer
$v^T < v^D$	kont.	Nein	Ja	Ja	str. \mathcal{NP} -schwer
$v^T < v^D$	disk.	Nein	Nein	Ja	str. \mathcal{NP} -schwer

Tabelle 1.1: Komplexität des Line-TSPwD verschiedener Szenarien

tisch beschreiben. Dabei handelt es sich um ein gemischt-ganzzahliges lineares Programm (engl. Mixed Integer Program, abgek. MIP) und einen Constraint Programming Ansatz (CP) für den diskreten Fall.

Abschließend wird in Kapitel 4 jeweils die Komplexität der verschiedenen Szenarien analysiert. Wie oben beschrieben, ist das Line-TSPTW ohne Zeitfenster einfach zu lösen. Durch das Hinzufügen einer Drohne, die schneller ist als der LKW und die an jedem Ort gestartet und abgeholt werden kann, wird das Problem \mathcal{NP} -schwer. Dies wird mithilfe einer Reduktion des Even-Odd-Partition-Problems gezeigt, welches ebenfalls \mathcal{NP} -schwer ist. Starten jedoch Drohne und LKW ganz links oder ganz rechts von der Linie, so ist das Problem wiederum polynomiell lösbar. Hierbei wird sich herausstellen, dass es sich um ein 1-Maschinen-Scheduling Problem handelt, wobei die Aufträge Prozesszeiten haben, die linear fallen. Diese beiden Ergebnisse sind in den Zeilen 5 und 6 der Tabelle 1.1 zusammengefasst. Ähnliches Vorgehen führt zu den weiteren Ergebnissen der Tabelle 1.1. Ein Großteil der Ergebnisse ist ebenfalls in einem parallel entstandenen und bisher unveröffentlichten Working Paper (Bock & Janiszczak, o. J.) zu finden.

Es sei darauf hingewiesen, dass in der vorliegenden Arbeit eine Vermischung von deutscher und englischer Sprache für einige Begriffe vorgenommen wird. Dies ist dadurch begründet, dass es sich um gängige Begriffe der Fachliteratur handelt.

Kapitel 2

Literaturüberblick

Eine Übersicht der Literatur für das TSP und das VRP bieten beispielsweise (Matai, Singh & Mittal, 2010) und (Golden, Raghavan & Wasil, 2008). (Chung, Sah & Lee, 2020) liefert einen Überblick der wissenschaftlichen Literatur im Bereich der 'drone-operation' (DO), also Probleme, die nicht an einen LKW geknüpft sind und 'drone-truck combined operations' (DTCO). Hierbei wird die Literatur in die Kategorien TSPD (Grundlage ist das TSP) und VRPD (Grundlage ist das VRP) aufgeteilt. Außerdem wird zwischen 'drone and truck as synchronized working units', 'Drone as primary and truck as secondary working units', 'Truck as primary and drone as secondary working units' und 'independent of both vehicles' unterschieden. Das Line-TSPwD ist also dem DTCO und insbesondere dem TSPD zuzuordnen, wobei beide Fahrzeuge gleich wichtig sind, d.h. sowohl die Drohne als auch der LKW bedienen Kunden und dies nicht unabhängig voneinander. Somit ist das Line-TSPwD als 'drone and truck as synchronized working units' einzustufen.

Ebenso geben (Otto, Agatz, Campbell, Golden & Pesch, 2018) und (Boysen, Fedtke & Schwerdfeger, 2021) einen Überblick über die Literatur zum Einsatz von Drohnen in der Logistik und Last-Mile-Delivery.

Erstmalig erwähnt wurden Drohnen im Zusammenhang mit dem TSP in (Murray & Chu, 2015). Hier wird das sogenannte Flying Sidekick TSP (FSTSP) und das Parallel Drone Scheduling TSP (PDSTSP) vorgestellt. Beim FSTSP liegt ein Depot und eine Menge von Kunden vor, wobei die Kunden dieser Menge bedient werden müssen. Hierbei gibt es eine Teilmenge T von Kunden, wobei diese Kunden ausschließlich vom LKW bedient werden können (z.B. aufgrund einer Kapazitätsbeschränkung der Drohne). Die Drohne kann genau einen Kunden bedienen, nachdem sie vom LKW aufgelesen wurde. Das Auflesen und das Starten der Drohne geschieht bei Kunden und nimmt jeweils eine konstante vorgegebene Zeit in Anspruch. Das Ziel ist eine frühest mögliche Ankunftszeit von Drohne und LKW am Depot, nachdem alle Kunden bedient wurden, entweder durch den LKW oder die Drohne. Hier müssen also die Drohnen-Tour und die LKW-Tour synchronisiert werden. Die Drohne kann nicht am selben Ort wieder aufgelesen werden, an dem sie gestartet wurde. Außerdem wird die Drohne nur an Orten gestartet und aufgelesen, an denen der LKW auch einen Kunden bedient, d.h. Kunden werden nicht mehrfach besucht. Die Autoren konstruieren für dieses Problem ein MIP und schlagen eine Heuristik vor. Beim PDSTSP hingegen arbeiten Drohnen und der LKW

unabhängig voneinander. Hier wird eine Aufteilung von Kunden gesucht, die nur durch Drohnen (in diesem Fall können es mehr als eine Drohne sein) oder nur vom LKW bedient werden. Hierbei pendeln die Drohnen zwischen dem Depot und den Kunden und sind somit unabhängig vom Fahrzeug. Für den LKW wird eine Tour konstruiert, welche im Depot startet und auch endet, bei der alle übrigen Kunden bedient werden. Auch für dieses Problem wird ein MIP und eine Heuristik beschrieben.

In (Dell'Amico, Montemanni & Novellani, 2021) wurden für das FSTSP eine Heuristik und ein Branch&Bound-Verfahren (B&B) präsentiert.

In (Agatz, Bouman & Schmidt, 2018) wird das TSP with Drone (TSP-D) vorgestellt, welches sehr ähnlich zum FSTSP ist. Im TSP-D nach (Agatz et al., 2018) ist es jedoch im Gegensatz zum FSTSP möglich, die Drohne an Kundenorten zu starten oder aufzulesen, obwohl der LKW die entsprechenden Kunden bereits vorher bedient hat. Dies ist beim FSTSP ausgeschlossen. Die Autoren stellen ein Integer Programming Modell (IP) vor sowie einige Heuristiken. Zusätzlich wird ein Ansatz der dynamischen Programmierung präsentiert, der eine optimale Partitionierung der Drohnen- und LKW-Kunden für eine fest vorgegebene Reihenfolge der Kunden ermittelt. In (Bouman, Agatz & Schmidt, 2018) wird ein Ansatz der Dynamischen Programmierung (DP) für das TSP-D konstruiert. Im Gegensatz zum TSP-D kann die Drohne in (Gonzalez-R, Canca, Andrade-Pineda, Calle & Leon-Blanco, 2020) mehrere Kunden nacheinander bedienen, ohne zum LKW zurückkehren zu müssen. Aufgrund der begrenzten Reichweite muss sie jedoch vom LKW aufgelesen werden, um weitere Kunden bedienen zu können. In (Tang et al., 2019) wird das TSP-D mit metrischen Distanzen analysiert. Darüber hinaus wird das Drone Routing Subproblem (DRS) definiert, bei dem eine LKW-Tour vorgegeben ist und eine Reihenfolge der Kunden ermittelt werden muss, in welcher die Drohne diese Kunden bedient. Die Autoren zeigen, dass das DRS \mathcal{NP} -schwer im strengen Sinne ist. Ein CP-Ansatz wird vorgestellt und mit anderen exakten Ansätzen verglichen.

Wie beim DRS in (Tang et al., 2019) liegt auch in (Boysen, Briskorn, Fedtke & Schwerdfeger, 2018) eine feste LKW-Tour vor. Hierbei können mehrere Drohnen die vorgegebenen Kunden bedienen. Jedoch werden die Drohnen immer nur bei LKW-Stopps gestartet oder aufgelesen. Die Autoren haben gezeigt, dass selbst im Fall, wenn nur eine Drohne involviert ist, das Problem mit beliebigen Distanzen \mathcal{NP} -schwer im strengen Sinne ist. Wird die Drohne am LKW-Stopp aufgelesen, an dem sie auch gestartet wurde, oder wird sie spätestens am nächsten LKW-Stopp aufgelesen, so lässt sich dieses Problem in Polynomialzeit lösen. Befinden sich auf dem LKW mehrere Drohnen, so ist das Problem selbst für einen LKW-Stopp ebenfalls \mathcal{NP} -schwer im strengen Sinne. Dies gilt ebenfalls unter der Voraussetzung, dass die Flugzeit vom k -ten LKW-Stopp zu Kunden j der Flugzeit in umgekehrter Richtung entspricht. Es werden zwei MIPs vorgestellt, die mithilfe eines Standardsolvers (Gurobi) getestet werden. Alle besprochenen Probleme setzen voraus, dass die Drohne nur an bestimmten Orten aufgelesen werden kann. In (Poikonen & Golden, 2020) hingegen ist das Aufladen der Drohne an jedem beliebigen Ort möglich. Betrachtet wird die euklidische Ebene \mathbb{R}^2 und eine beschränkte Reichweite der Drohne. Dieses Problem ist \mathcal{NP} -schwer im strengen Sinne. In der vorliegenden Arbeit wird gezeigt, dass das Problem unter der Beschränkung, dass sich beide Fahrzeuge nur auf einer Linie, also in \mathbb{R} bewegen \mathcal{NP} -schwer ist.

Kapitel 3

Modelle

3.1 Was ist eine Drohne?

Die Internationale Zivilluftfahrtorganisation (ICAO) definiert ein UAS (Unmanned Aircraft System) als ein Fluggerät mit all seinen zugehörigen Elementen, das keinen Piloten an Board hat. Ein UA (Unmanned Aircraft) ist ein Fluggerät ohne Piloten an Bord (International Civil Aviation Organization, 2011). In der Literatur wird auch häufig die Abkürzung UAV (Unmanned Aircraft Vehicle) genutzt, wie zum Beispiel in (Murray & Chu, 2015). Eine Klassifizierung von UAS nach verschiedenen Eigenschaften, wie z.B. dem Gewicht oder der Reichweite ist in (Dalamagkidis, 2015) zu finden.

In der vorliegenden Arbeit beschreibt der Begriff *Drohne* durchgängig ein Flugobjekt, das autonom Pakete ausliefern kann. Die ingenieurtechnische Umsetzung wird hierbei in dieser Arbeit nicht behandelt. Es wird nur das Problem aus theoretischer Sicht analysiert.

Als Beispiel für Drohnen können die des Unternehmens Zipline genannt werden. Diese haben eine Reichweite von 160 km und eine Ladekapazität von 1,3 kg. Darüber hinaus ist eine Geschwindigkeit von bis zu 128 km/h möglich, wobei eine Reisegeschwindigkeit von 101 km/h erreicht wird (Zipline, o.J.). Die Lieferdrohne *Wingcopter 198* des deutschen Unternehmens Wingcopter hat bei einer maximalen Zuladung von 6 kg eine Reichweite von 75 km (bis maximal 110 km bei geringerer Belastung) und hat eine Geschwindigkeit von 150km/h (Koenen, 2021).

3.2 Gemischt-ganzzahliges lineares Programm

3.2.1 Parameter

Gegeben sei eine Line-TSPwD Instanz mit n Kunden 1 bis n . Das bedeutet, dass sich diese Kunden auf einer Linie, zum Beispiel auf einer Straße, oder entlang eines Flusses befinden. Ein LKW und eine Drohne bedienen die Kunden, wobei der LKW eine konstante Geschwindigkeit von v^T und die Drohne eine konstante Geschwindigkeit von v^D haben. Dabei befinden sich beide zu Beginn beim Kunden $i^* \in \{1, \dots, n\}$ (siehe Abbildung 3.1). Nachdem ein Kunde durch

die Drohne bedient wurde, muss diese zunächst zum LKW zurückfliegen, bevor sie den nächsten Kunden bedienen kann. Der Kunde i befindet sich an Position p_i . Es wird zunächst für $i < j$ aus der Kundenmenge die Eigenschaft $p_i < p_j$ vorausgesetzt. Damit resultiert zwischen Kunden i und Kunden j mit $i < j$ ein Abstand von $p_j - p_i$. Jeder Kunde muss besucht werden. Außerdem ist jeder Kunde mit einem Zeitfenster verknüpft, in welchem das Bedienen des Kunden möglich ist. Dabei ist für Kunde i Der Zeitpunkt r_i (Release Date) der Beginn des Zeitfensters und d_i (Deadline) das Ende des Zeitfensters. Der Zeitaufwand, um einen Kunden zu bedienen, wird als vernachlässigbar, also als 0 vorausgesetzt. Es gibt eine Teilmenge T aller Kunden, wobei die Kunden dieser Teilmenge aufgrund von Kapazitätsbeschränkungen der Drohne nicht von ihr bedient werden können, sondern vom LKW bedient werden müssen. Es wird zwischen zwei Zielfunktionen unterschieden: die Minimierung des Zeitpunktes des Aufeinandertreffens von Drohne und LKW, nachdem alle Kunden besucht wurden und die Minimierung des Zeitpunktes, zu dem der letzte Kunde besucht wurde.

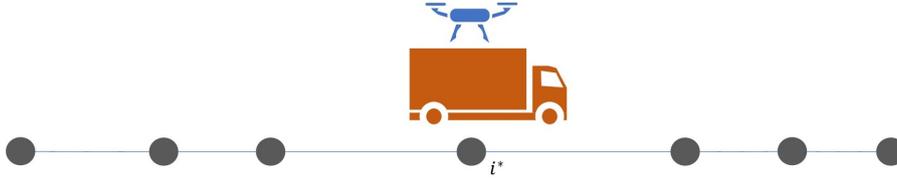


Abbildung 3.1: Line-TSPwD Instanz

Zusammengefasst liegen also die folgenden Parameter vor:

n	Anzahl der Kunden
i^*	Startkunde (Hier befinden sich Drohne und LKW zum Zeitpunkt $t = 0$)
T	Menge der Kunden, die nur durch den LKW bedient werden können
p_i	Position von Kunde i
r_i	Release Date von Kunde i
d_i	Deadline von Kunde i
v^D	Geschwindigkeit der Drohne
v^T	Geschwindigkeit des LKWs

Tabelle 3.1: Parameter einer Line-TSPwD Instanz

3.2.2 Variablen

Kunde i wird zum Zeitpunkt t_i bedient. Wird Kunde i durch die Drohne bedient, müssen sowohl der Trennungsort, der mit r_i^s bezeichnet wird, als auch der Trennungzeitpunkt, welcher mit t_i^s bezeichnet wird, festgelegt werden. Der Begegnungsort nach dem Bedienen des Kunden i ist r_i^m . Im *diskreten Fall* sind die Begegnungs- und Trennungsorte auf eine diskrete Menge eingeschränkt, nämlich $r_i^m, r_i^s \in \{p_1, \dots, p_n\}$. Hingegen ist im *kontinuierlichen Fall* das Starten und das

Auflesen der Drohne an jedem Ort auf der Linie möglich. Der Begegnungszeitpunkt wird durch die Variable t_i^m dargestellt. Außerdem existieren für jeden Kunden i die folgenden Binärvariablen:

$$dr_i = \begin{cases} 1 & i \text{ wird durch Drohne bedient,} \\ 0 & \text{sonst} \end{cases}$$

$$tr_i = \begin{cases} 1 & i \text{ wird durch LKW bedient,} \\ 0 & \text{sonst} \end{cases}$$

Nachdem die Drohne einen Kunden bedient hat, ist ihr nächstes Ziel die Begegnung mit dem LKW, um den nächsten Kunden bedienen zu können. Somit kann man eine Lösung als eine Hintereinanderausführung der Operationen "Kunde bedienen", "Drohne zum Trennungsort bringen" und "Drohne am Begegnungsort abholen" auffassen. Dementsprechend sind maximal $2n$ Operationen notwendig. Dieser Fall tritt ein, wenn die Drohne alle Kunden bedient, denn dann muss der LKW die Drohne jeweils zum Trennungsort bringen und beim Begegnungsort einsammeln. Die Operationen "Kunde bedienen" wird als *LKW-Operation* bezeichnet und die Operationen "Drohne am Begegnungsort abholen" und "Drohne zum Trennungsort bringen" werden als *Drohnen-Operationen* bezeichnet. Die folgenden Variablen legen die Reihenfolge dieser Operationen fest:

$$x_{i,k} = \begin{cases} 1 & \text{Operation } i \text{ ist die } k\text{-te Operation,} \\ 0 & \text{sonst} \end{cases}$$

Dabei sind $i \in \{1, \dots, 3n\}$ und $k \in \{1, \dots, 2n\}$. Die Elemente $i \in \{1, \dots, n\}$ repräsentieren die Operationen "LKW bedient Kunde i ", die Elemente $i \in \{n+1, \dots, 2n\}$ repräsentieren die Operationen "LKW bringt Drohne zum Trennungsort, damit Drohne Kunde $i-n$ bedient" und die Elemente $j \in \{2n+1, \dots, 3n\}$ repräsentieren die Operationen "LKW holt Drohne vom Begegnungsort ab, nachdem Drohne Kunde $j-2n$ besucht hat". Die Elemente $k \in \{1, \dots, 2n\}$ geben die Positionen der Reihenfolge der Operationen an. Zusammengefasst liegen also die folgenden Variablen vor:

t_i	Fertigstellungszeitpunkt von Kunde i
r_i^s	Trennungsort von Drohne und LKW, damit Drohne Kunde i bedient
r_i^m	Begegnungsort von Drohne und LKW, nachdem Drohne Kunde i bedient hat
t_i^s	Trennungszeitpunkt von Drohne und LKW, damit Drohne Kunde i bedient
t_i^m	Begegnungszeitpunkt von Drohne und LKW, nachdem Drohne Kunde i bedient hat
dr_i	Binärvariable die entscheidet, ob Kunde i durch die Drohne bedient wird
tr_i	Binärvariable die entscheidet, ob Kunde i durch den LKW bedient wird
$x_{i,k}$	Binärvariable die entscheidet, ob eine Operation i an Position k in der Sequenz durchgeführt wird

Tabelle 3.2: Variablen des Modells

3.2.3 Modell

Das Line-TSPwD mit n Kunden lässt sich als gemischt-ganzzahliges Programm wie folgt darstellen. Sei M eine große Zahl, sodass die Restriktionen, die diesen Parameter beinhalten entweder relevant sind oder nicht. M ist also derart gewählt, dass die jeweiligen Ungleichungen immer erfüllt sind, wenn die zugehörigen Binärvariablen M nicht eliminieren. Außerdem wird vorausgesetzt, dass $v^D, v^T > 0$ ist. Seien $N := \{1, \dots, n\}$, $2N := \{1, \dots, 2n\}$ und $3N := \{1, \dots, 3n\}$.

$$\min \quad z \quad (3.1)$$

$$\text{sodass} \quad z \geq t_i \quad \forall i \in N \quad (3.2)$$

$$z \geq t_i^m \quad \forall i \in N \quad (3.3)$$

$$t_i \geq r_i \quad \forall i \in N \quad (3.4)$$

$$t_i \leq d_i \quad \forall i \in N \quad (3.5)$$

$$tr_i = 1 \quad \forall i \in T \quad (3.6)$$

$$x_{i^*,1} = 1 \quad (3.7)$$

$$x_{i^*+2n,2} = 1 \quad (3.8)$$

$$r_{i^*}^m = p_{i^*} \quad (3.9)$$

$$t_{i^*}^m = 0 \quad (3.10)$$

$$t_{i^*} = 0 \quad (3.11)$$

$$\sum_{k=1}^{2n} x_{i,k} \leq 1 \quad \forall i \in 3N \quad (3.12)$$

$$\sum_{i=1}^{3n} x_{i,k} \leq 1 \quad \forall k \in 2N \quad (3.13)$$

$$M \cdot \sum_{i=1}^{3n} x_{i,k} \geq \sum_{j=1}^{3n} \sum_{l=k+1}^{2n} x_{j,l} \quad \forall k \in 2N \quad (3.14)$$

$$\sum_{\substack{i=1 \\ i \neq i^*}}^n tr_i + 2 \sum_{\substack{i=1 \\ i \neq i^*}}^n dr_i = \sum_{i=1}^{3n} \sum_{k=3}^{2n} x_{i,k} \quad (3.15)$$

$$tr_i = \sum_{k=1}^{2n} x_{i,k} \quad \forall i \in N \quad (3.16)$$

$$dr_i = \sum_{k=1}^{2n} x_{i+n,k} \quad \forall i \in N \quad (3.17)$$

$$dr_i = \sum_{k=1}^{2n} x_{i+2n,k} \quad \forall i \in N \quad (3.18)$$

$$tr_i + dr_i = 1 \quad \forall i \in N \quad (3.19)$$

$$2M - (x_{i,k} + x_{j,k+1})M + t_j \geq t_i + \frac{1}{vT}(|p_i - p_j|) \quad \forall i, j \in N, k \in 2N \setminus \{2n\} \quad (3.20)$$

$$2M - (x_{i+n,k} + x_{j,k+1})M + t_j \geq t_i^s + \frac{1}{vT}(|r_i^s - p_j|) \quad \forall i, j \in N, k \in 2N \setminus \{2n\} \quad (3.21)$$

$$2M - (x_{i+2n,k} + x_{j,k+1})M + t_j \geq t_i^m + \frac{1}{vT}(|r_i^m - p_j|) \quad \forall i, j \in N, k \in 2N \setminus \{2n\} \quad (3.22)$$

$$2M - (x_{j,k} + x_{i+n,k+1})M + t_i^s \geq t_j + \frac{1}{vT}(|r_i^s - p_j|) \quad \forall i, j \in N, k \in 2N \setminus \{2n\} \quad (3.23)$$

$$2M - (x_{j,k} + x_{i+2n,k+1})M + t_i^m \geq t_j + \frac{1}{vT}(|r_i^m - p_j|) \quad \forall i, j \in N, k \in 2N \setminus \{2n\} \quad (3.24)$$

$$2M - (x_{i+2n,k} + x_{j+n,k+1})M + t_j^s \geq t_i^m + \frac{1}{vT}(|r_i^m - r_j^s|) \quad \forall i, j \in N, k \in 2N \setminus \{2n\} \quad (3.25)$$

$$2M - (x_{i+n,k} + x_{i+2n,k+1})M + t_i^m \geq t_i + \frac{1}{vD}(|p_i - r_i^m|) \quad \forall i, j \in N, k \in 2N \setminus \{2n\} \quad (3.26)$$

$$2M - (x_{i+n,k} + x_{i+2n,k+1})M + t_i^m \geq t_i^s + \frac{1}{vT}(|r_i^s - r_i^m|) \quad \forall i, j \in N, k \in 2N \setminus \{2n\} \quad (3.27)$$

$$(1 - dr_j)M + t_j \geq t_j^s + \frac{1}{vD}(|r_j^s - p_j|) \quad \forall j \in N \quad (3.28)$$

$$(1 - dr_j)M + t_j^m \geq t_j + \frac{1}{vD}(|r_j^m - p_j|) \quad \forall j \in N \quad (3.29)$$

$$\sum_{k=1}^{2n} x_{i+n,k} \cdot k \leq \sum_{k=1}^{2n} x_{i+2n,k} \cdot k \quad \forall i \in N \quad (3.30)$$

$$2M - (x_{i+n,k} + x_{i+2n,j})M + \sum_{l=k+1}^{j-1} \sum_{m=1}^n x_{m,l} \geq j - k - 1 \quad \forall i \in N, k, j \in 2N \quad (3.31)$$

$$\begin{array}{ll} x_{i,j} \in \{0, 1\} & \forall i, j \in 3N \\ tr_i \in \{0, 1\} & \forall i \in N \\ dr_i \in \{0, 1\} & \forall i \in N \\ r_i^m \geq 0 & \forall i \in N \\ r_i^s \geq 0 & \forall i \in N \\ t_i^m \geq 0 & \forall i \in N \end{array}$$

$$\begin{aligned} t_i^s &\geq 0 & \forall i \in N \\ z &\geq 0 \end{aligned}$$

Falls das Ziel die Minimierung des Zeitpunkts des zuletzt besuchten Kunden ist, so ist Restriktion (3.3) zu entfernen. In dem Fall, dass das Ziel die Minimierung des Begegnungszeitpunkts von Drohne und LKW ist, nachdem alle Kunden rechtzeitig bedient wurden, so ist Restriktion (3.3) relevant. Die Restriktionen (3.4) und (3.5) führen zum Einhalten der Zeitfenster, und durch (3.6) wird gewährleistet, dass Kunden, welche die Drohne nicht bedienen kann, durch den LKW bedient werden. Durch (3.7) bis (3.11) wird sichergestellt, dass LKW und Drohne sich zum Zeitpunkt 0 bei i^* befinden.

Jede Operation wird maximal einer Position der Sequenz zugeordnet. Diese Beschränkung realisiert Restriktion (3.12). Mit (3.13) ist die k -te Operation eindeutig, falls diese existiert. In der Sequenz sind nur die letzten Positionen nicht belegt. Sobald die k -te Operation eine leere Operation ist, also keine LKW-Operation stattfindet, sind auch alle darauffolgenden Positionen in der Sequenz leere Operationen. Hierfür wird Restriktion (3.14) verwendet. Mit (3.15) sind genau so viele Positionen in der Sequenz mit nicht-leeren Operationen besetzt, wie es LKW-Operationen gibt. Die Beschränkungen (3.15) bis (3.18) halten fest, welche Kunden durch den LKW und welche Kunden durch die Drohne bedient werden. Wird die Drohne gestartet, um einen Kunden zu bedienen, so wird sichergestellt, dass die Drohne auch wieder aufgelesen wird. Bedingung (3.19) gewährleistet, dass jeder Kunde bedient wird.

Werden Kunde i und j durch den LKW bedient und j direkt nach Kunde i , so kann der LKW frühestens ab der Fertigstellung von i zu j fahren und hierbei die Strecke zwischen p_i und p_j zurücklegen. Hierfür wird Bedingung (3.20) herangezogen, denn sind $x_{i,k} = x_{j,k+1} = 1$, so folgt die Nebenbedingung $t_j \geq t_i + (|p_i - p_j|)/v^T$. Ebenso für die Fertigstellungszeitpunkte, Begegnungszeitpunkte und Trennungszeitpunkte realisieren (3.21) bis (3.29) für alle möglichen Kombinationen der anderen Operationen die jeweiligen Beschränkungen. Hierbei ist anzumerken, dass sich das obige Modell in ein Modell überführen lässt, dessen Beschränkungen ein lineares Ungleichungssystem bilden, da die folgende Äquivalenz gilt:

$$x \geq |y| \Leftrightarrow x \geq y \wedge x \geq -y$$

Durch (3.30) wird die Drohne später eingesammelt, als sie gestartet wird, um Kunde i zu bedienen. Wird die Drohne gestartet, können bis zum Einsammeln der Drohne nur Operationen der Form "Kunde wird durch LKW bedient" stattfinden. Dies garantiert (3.31).

Für das gegebene Modell lässt sich mithilfe der übrigen Parameter die Zahl M bestimmen.

$$M = 2n + \frac{n \cdot |p_n - p_1|}{v^T} + \max_{i \in N} \{r_i\}$$

Der Zielfunktionswert einer Lösung für das Line-TSPWT bzw. einer Lösung für das Line-TSPwD unter der Bedingung $T = \{1, \dots, n\}$ ist eine obere Schranke für das Line-TSPwD. Wenn der LKW also alle Kunden bedienen muss, wird seine Fahrzeit nach oben durch $\frac{n|p_n - p_1|}{v^T}$ begrenzt, denn um den nächsten Kunden anzufahren, ist die Strecke nie länger als $|p_n - p_1|$. Hinzu kommen Wartezeiten, die aber $\max\{r_i\}$ nicht überschreiten.

Falls eine zulässige Lösung existiert, so ist der optimale Zielfunktionswert nicht

größer als M . Damit sind Restriktionen (3.20) bis (3.29) immer erfüllt, falls den entsprechenden Binärvariablen Werte zugewiesen werden, sodass M nicht eliminiert wird.

Die rechte Seite der Restriktion (3.14) ist stets kleiner oder gleich $2n$. Ebenso erfüllt M die gewünschte Eigenschaft für Restriktion (3.31), denn die rechte Seite der Restriktion übersteigt $2n$ nicht.

Werden die Domänen $r_i^m \geq 0 \forall i \in \{1, \dots, n\}$ und $r_i^s \geq 0 \forall i \in \{1, \dots, n\}$ ersetzt durch

$$r_i^m \in \{p_j \mid j \in \{1, \dots, n\}\} \text{ und } r_i^s \in \{p_j \mid j \in \{1, \dots, n\}\},$$

so liegt die diskrete Variante vor.

An dieser Stelle sei angemerkt, dass durch eine geringfügige Modifikation des Modells eine Rückkehr des LKWs zum Startkunden i^* realisiert werden kann. Dies ist in der Literatur eine häufige Forderung (z.B. (Murray & Chu, 2015), (Agatz et al., 2018), (Tang et al., 2019)). Das Modell lässt sich in ein neues umformulieren, indem die Variablen t_{Depot} , tr_{Depot} , $x_{Depot,k}$ für $k \in \{1, \dots, 2n\}$ sowie der Parameter $p_{Depot} = p_{i^*}$ hinzugefügt werden. Darüber hinaus müssen die folgenden Restriktionen zusätzlich erfüllt sein:

$$tr_{Depot} = 1 \tag{3.32}$$

$$\sum_{k=1}^{2n} k \cdot x_{Depot,k} \geq \sum_{j=1}^{2n} j \cdot x_{l,j} \quad \forall l \in \{1, \dots, 3n\} \tag{3.33}$$

Nebenbedingung (3.32) führt dazu, dass der LKW den Startort erneut besucht, wobei (3.33) sicherstellt, dass dieser Ort zuletzt besucht wird.

Um Lösungen übersichtlicher zu gestalten, wird der Vektor $a \in \{1, \dots, 3n\}^{2n}$ mit $a_i := \sum_{j=1}^n j \cdot x_{j,i}$ definiert. Dabei entspricht der i -te Eintrag des Vektors der i -ten Operation.

Definition 3.2.1. Für das Line-TSP with Drone Problem mit n Kunden heißt ein Vektor $a = (a_1, \dots, a_{2n}) \in \{1, \dots, 3n\}^{2n}$ partiell zulässig, wenn die folgenden Eigenschaften erfüllt sind:

- $a_1 = i^*$ und $a_2 = i^* + 2 \cdot n$
- Es existieren Mengen A und B mit $A \dot{\cup} B = \{1, \dots, n\} \setminus \{i^*\}$, sodass

$$\{a_3, \dots, a_{2n}\} = \{i \mid i \in A\} \cup \{i + n \mid i \in B\} \cup \{i + 2n \mid i \in B\}$$

oder

$$\{a_3, \dots, a_{2n}\} = \{i \mid i \in A\} \cup \{i + n \mid i \in B\} \cup \{i + 2n \mid i \in B\} \cup \{0\}$$

- Es gilt $T \subseteq A$.
- Existiert $l \in \{1, \dots, 2n\}$ mit $a_l = 0$, so ist $a_m = 0$ für $m \in \{l + 1, \dots, 2n\}$
- Für alle $i, j \in \{1, \dots, 2n\}$ mit $i \neq j$ und $a_i \neq 0$ gilt $a_i \neq a_j$
- Ist $a_k = i + n$ und $a_l = i + 2n$ mit $i \in \{1, \dots, n\}$, so ist $k < l$ und $a_m \in \{1, \dots, n\}$ für alle $m \in \{k + 1, \dots, l - 1\}$

Ein partiell zulässiger Vektor erfüllt somit die Bedingungen, dass alle Kunden bedient werden und bei i^* begonnen wird. Außerdem wird die Drohne nur gestartet, nachdem der LKW sie aufgelesen hat. Dabei wird berücksichtigt, dass die Drohne gestartet wird, um einen ausgezeichneten Kunden zu bedienen und anschließend aufgelesen wird, nachdem sie diesen Kunden bedient hat.

Diese Eigenschaften werden im obigen Modell durch die Restriktionen (3.6), (3.7), (3.30), (3.31) sowie (3.12) bis (3.19) realisiert.

Ist ein partiell zulässiger Vektor $a = (a_1, \dots, a_{2 \cdot n}) \in \{1, \dots, 3 \cdot n\}^{2 \cdot n}$ für ein kontinuierliches Problem gegeben, so lassen sich in Polynomialzeit die optimalen Begegnungsorte und Begegnungszeiten ermitteln. Dazu ist das folgende lineare Programm zu lösen:

$$\min \quad z$$

sodass

$$z \geq t_i \quad \forall i \in N \quad (3.34)$$

$$z \geq t_i^m \quad \forall i \in N \quad (3.35)$$

$$t_i \geq r_i \quad \forall i \in N \quad (3.36)$$

$$t_i \leq d_i \quad \forall i \in N \quad (3.37)$$

$$r_{i^*}^m = p_{i^*} \quad (3.38)$$

$$t_{i^*}^m = 0 \quad (3.39)$$

$$t_{i^*} = 0 \quad (3.40)$$

$$t_{a_{i+1}} \geq \frac{1}{vT} (|p_{a_{i+1}} - p_{a_i}|) + t_{a_i} \quad \forall i \in N : a_i, a_{i+1} \in \{1, \dots, n\} \quad (3.41)$$

$$t_{a_{i+1}} \geq \frac{1}{vT} (|p_{a_{i+1}} - r_{a_i-2 \cdot n}^m|) + t_{a_i-2 \cdot n}^m \quad (3.42)$$

$$\forall i \in N : a_{i+1} \in \{1, \dots, n\}, a_i \in \{2 \cdot n + 1, \dots, 3 \cdot n\}$$

$$t_{a_i-n} \geq \frac{1}{vD} (|p_{a_i} - r_{a_i-n}^s|) + t_{a_i-n}^s \quad \forall i \in N : a_i \in \{n+1, \dots, 2 \cdot n\} \quad (3.43)$$

$$t_{a_i-2 \cdot n}^m \geq \frac{1}{vD} (|p_{a_i-2 \cdot n} - r_{a_i-2 \cdot n}^m|) + t_{a_i-2 \cdot n} \quad (3.44)$$

$$\forall i \in N : a_i \in \{2 \cdot n + 1, \dots, 3 \cdot n\}$$

$$t_{a_{i+1}-2 \cdot n}^m \geq \frac{1}{vT} (|p_{a_i} - r_{a_{i+1}-2 \cdot n}^m|) + t_{a_i} \quad (3.45)$$

$$\forall i \in N : a_{i+1} \in \{2 \cdot n + 1, \dots, 3 \cdot n\}, a_i \in \{1, \dots, n\}$$

$$t_{a_{i+1}-n}^s \geq \frac{1}{vT} (|p_{a_i} - r_{a_{i+1}-n}^s|) + t_{a_i} \quad (3.46)$$

$$\forall i \in N : a_{i+1} \in \{n+1, \dots, 2 \cdot n\}, a_i \in \{1, \dots, n\}$$

$$t_{a_{i+1}-n}^s \geq \frac{1}{vT} (|r_{a_i-2 \cdot n}^m - r_{a_{i+1}-n}^s|) + t_{a_i-2 \cdot n}^m \quad (3.47)$$

$$\forall i \in N : a_{i+1} \in \{n+1, \dots, 2 \cdot n\}, a_i \in \{2 \cdot n + 1, \dots, 3 \cdot n\}$$

$$t_{a_{i+1}-n}^m \geq \frac{1}{v^T} (|r_{a_i-n}^s - r_{a_{i+1}-2n}^m|) + t_{a_i-n}^s \quad (3.48)$$

$$\forall i \in N : a_i \in \{n+1, \dots, 2 \cdot n\}, a_{i+1} \in \{2 \cdot n+1, \dots, 3 \cdot n\}$$

$$t_{a_{i+1}} \geq \frac{1}{v^T} (|p_{a_{i+1}} - r_{a_i-n}^s|) + t_{a_i-n}^s \quad (3.49)$$

$$\forall i \in N : a_{i+1} \in \{1, \dots, n\}, a_i \in \{n+1, \dots, 2 \cdot n\}$$

$$\begin{aligned} t_i &\geq 0 & \forall i \in N \\ t_i^m &\geq 0 & \forall i \in N \\ t_i^s &\geq 0 & \forall i \in N \\ r_i^m &\geq 0 & \forall i \in N \\ r_i^s &\geq 0 & \forall i \in N \end{aligned}$$

Durch das Entfernen der Restriktion (3.35) ist das Ziel die Minimierung von $\max\{t_i\}$.

Insgesamt liegt die Anzahl der Restriktionen und Variablen in $\mathcal{O}(n)$, denn (3.41), (3.42), (3.46), (3.47), (3.48) und (3.49) sind aufsummiert maximal n Restriktionen. Es gibt jeweils maximal n Restriktionen vom Typ (3.34), (3.35), (3.43), (3.44) und (3.45). Lineare Programme sind mit einigen Verfahren in Polynomialzeit lösbar, wie zum Beispiel mit dem Innere-Punkte-Verfahren in (Karmarkar, 1984).

Beispiel 3.2.2. Sei die folgende partiell zulässige Lösung für eine kontinuierliche Line-TSPwD Instanz mit $n = 10$ und $i^* = 4$ gegeben:

$$a = (4, 24, 7, 11, 5, 21, 12, 9, 8, 6, 22, 13, 23, 10, 0, 0, 0, 0, 0, 0).$$

Diese ist tatsächlich partiell zulässig, da $a_1 = i^* = 4$ und $a_2 = 24 = i^* + 2n$. Außerdem ist $A = \{5, 6, 7, 8, 9, 10\}$ und $B = \{1, 2, 3\}$. Es gilt $\min\{i \in \{1, \dots, 2n\} | a_i = 0\} = 15$ und $a_i = 0$ für alle $15 \leq i \leq 20$. Darüber hinaus ist $a_i \neq a_j$ für alle $i \neq j$ mit $a_i \neq 0$.

Außerdem ist gewährleistet, dass für alle $i \in \{1, \dots, n\}$, für die es ein $k \in \{1, \dots, n\}$ gibt mit $a_k = i + n$, auch ein $l > k$ existiert mit $a_l = i + 2n$. Liegen nun keine Deadlines vor, so ist diese Reihenfolgezuordnung zulässig, und eine optimale Lösung bezüglich dieser Festlegung lässt sich mit dem oben definierten linearen Programm finden. Hierbei sind die Geschwindigkeiten $v^D = 5$ und $v^T = 2, 5$, und die Positionen der Kunden sind der nachfolgenden Tabelle zu entnehmen, welche ebenso die Werte einer optimalen Lösung des vorgegebenen Vektors a enthält:

Kunde i	1	2	3	4	5	6	7	8	9	10
Position	0	30	40	70	90	140	170	300	340	360
t_i	74	230	272	0	72	252	40	188	172	340
r_i^m	190	140	360	-	-	-	-	-	-	-
t_i^m	112	252	340	-	-	-	-	-	-	-
r_i^s	170	333.3	140	-	-	-	-	-	-	-
t_i^s	40	169.3	252	-	-	-	-	-	-	-

Alle Beispielinstanzen in dieser Arbeit wurden mit dem Solver IBM ILOG CPLEX Optimization Studio gelöst. Es ist anzumerken, dass diese Lösung nicht

optimal für die gegebene Line-TSPwD Instanz ohne festgelegte partielle Lösung ist. Die Lösung lässt sich mithilfe eines Zeit-Ort-Diagramms wie folgt darstellen:

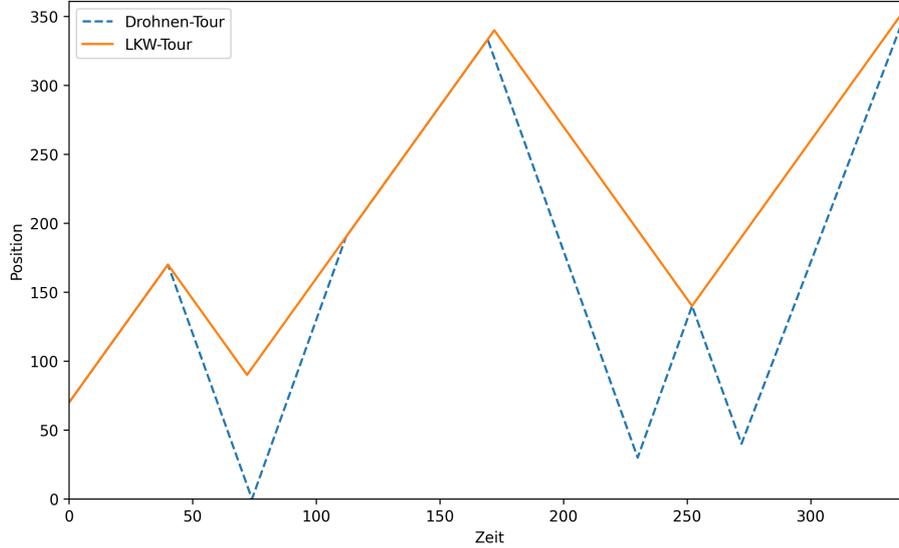


Abbildung 3.2: Lösung für gegebene Reihenfolgefestlegung a

3.2.4 Anzahl aller partiell zulässigen Lösungen

Im Folgenden wird eine Formel hergeleitet, welche die Anzahl aller möglichen partiell zulässigen Lösungen berechnet. Dazu wird zunächst der Fall analysiert, wie viele Möglichkeiten es gibt, die Operationen "Drohne wird gestartet" und "Drohne wird aufgelesen" zu wählen, wobei die jeweilige Menge der Kunden (mit gegebener Reihenfolge) festgelegt ist, welche durch den LKW bzw. durch die Drohne bedient werden. Dies soll zunächst anhand eines kleinen Beispiels gezeigt werden.

Beispiel 3.2.3. Seien $m = 2$ Kunden, die durch den LKW bedient werden und $k = 2$ Kunden, die durch die Drohne bedient werden, gegeben. Darüber hinaus wird eine Reihenfolge bezüglich der LKW-Operationen und der Drohnen-Operationen festgelegt.

Seien $(\sigma^T(1), \sigma^T(2))$ die Reihenfolgefestlegung der Kunden, die durch den LKW bedient werden und $(\sigma^D(1), \sigma^D(2))$ die Reihenfolgefestlegung der Kunden, die durch die Drohne bedient werden. Es gilt also $\{\sigma^T(1), \sigma^T(2), \sigma^D(1), \sigma^D(2)\} = \{1, 2, 3, 4, 5\} \setminus \{i^*\}$. Nun werden alle Möglichkeiten gezählt, wie sich die Drohnen-Operationen zwischen den LKW-Operationen integrieren lassen, sodass ein partiell zulässiger Vektor resultiert.

Um die Position der ersten Operation "Drohne wird gestartet" zu wählen, gibt es 3 Möglichkeiten, nämlich vor "LKW bedient $\sigma^T(1)$ ", nach "LKW bedient $\sigma^T(1)$ " und nach "LKW bedient $\sigma^T(2)$ ". Dies entspricht der zweiten Ebene des Baums in Abbildung 3.3. Wird die Drohne vor "LKW bedient $\sigma^T(1)$ " gestartet, so gibt es für das Auflesen der Drohne 3 Möglichkeiten, nämlich vor "LKW bedient

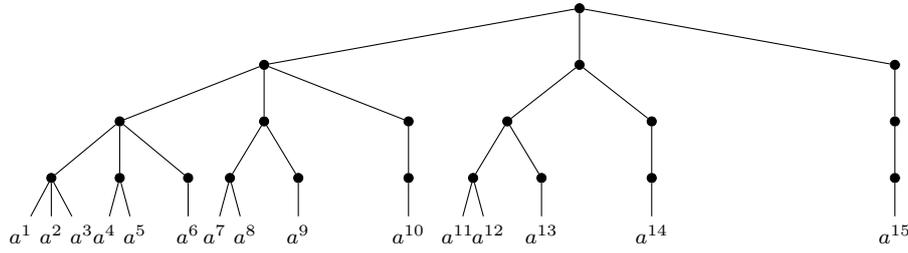


Abbildung 3.3: Alle Reihenfolgefestlegungen von LKW-Operationen mit fester Reihenfolge zweier Kunden, die durch die Drohne bedient werden und zweier Kunden, die durch den LKW bedient werden

$\sigma^T(1)$ “, nach *”LKW bedient $\sigma^T(1)$ “* und nach *”LKW bedient $\sigma^T(2)$ “*. Dies entspricht den Verzweigungen des ersten Knotens der zweiten Ebene des Baums. Wird die Drohne nach *”LKW bedient $\sigma^T(1)$ “* gestartet, so reduziert sich die Anzahl der Möglichkeiten des Einsammelns der Drohne auf das Einsammeln nach *”LKW bedient $\sigma^T(1)$ “* und nach *”LKW bedient $\sigma^T(2)$ “*, es gibt also nur noch 2 Möglichkeiten. Wird die Drohne nach *”LKW bedient $\sigma^T(2)$ “* gestartet, so gibt es nur eine Möglichkeit für das Auflesen der Drohne, nämlich nach *”LKW bedient $\sigma^T(2)$ “*. In der Summe gibt es also 6 Möglichkeiten die Drohne zu starten und einzusammeln, wenn es zwei feste Kunden gibt, die durch den LKW bedient werden (dritte Ebene in Abbildung 3.3). Im nächsten Schritt wird nun die Anzahl aller Möglichkeiten ermittelt, die Drohne zu starten um Kunde $\sigma^D(2)$ zu bedienen und anschließend wieder einzusammeln. Für die erste dieser beiden Operationen liegen nun 3 Möglichkeiten vor, falls die Drohnen-Operationen $\sigma^D(1) + 5$ und $\sigma^D(1) + 10$ vor *”LKW bedient $\sigma^T(1)$ “* stattfinden:

$$\begin{aligned} & (i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^T(1), \sigma^T(2)) \\ & (i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^T(1), \sigma^T(2) + 5, \sigma^D(2)) \\ & (i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^T(1), \sigma^T(2), \sigma^D(2) + 5) \end{aligned}$$

Diese Teillösungen werden in den drei linken Knoten der vorletzten Ebene in Abbildung 3.3 repräsentiert. Auf diese Weise erhält man insgesamt 15 Möglichkeiten, die nachfolgend angegeben werden:

a^1	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^D(2) + 10, \sigma^T(1), \sigma^T(2))$
a^2	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^T(1), \sigma^D(2) + 10, \sigma^T(2))$
a^3	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^T(1), \sigma^T(2), \sigma^D(2) + 10)$
a^4	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^T(1), \sigma^D(2) + 5, \sigma^D(2) + 10, \sigma^T(2))$
a^5	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^T(1), \sigma^D(2) + 5, \sigma^T(2), \sigma^D(2) + 10)$
a^6	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^T(1), \sigma^T(2), \sigma^D(2) + 5, \sigma^D(2) + 10)$
a^7	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^T(1), \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^D(2) + 10, \sigma^T(2))$
a^8	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^T(1), \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^T(2), \sigma^D(2) + 10)$
a^9	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^T(1), \sigma^D(1) + 10, \sigma^T(2), \sigma^D(2) + 5, \sigma^D(2) + 10)$
a^{10}	$(i^*, i^* + 10, \sigma^D(1) + 5, \sigma^T(1), \sigma^T(2), \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^D(2) + 10)$
a^{11}	$(i^*, i^* + 10, \sigma^T(1), \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^D(2) + 10, \sigma^T(2))$
a^{12}	$(i^*, i^* + 10, \sigma^T(1), \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^T(2), \sigma^D(2) + 10)$
a^{13}	$(i^*, i^* + 10, \sigma^T(1), \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^T(2), \sigma^D(2) + 5, \sigma^D(2) + 10)$
a^{14}	$(i^*, i^* + 10, \sigma^T(1), \sigma^D(1) + 5, \sigma^T(2), \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^D(2) + 10)$
a^{15}	$(i^*, i^* + 10, \sigma^T(1), \sigma^T(2), \sigma^D(1) + 5, \sigma^D(1) + 10, \sigma^D(2) + 5, \sigma^D(2) + 10)$

Anmerkung 3.2.4. Eine Multimenge M kann ein und dasselbe Objekt mehrfach enthalten, z.B. $M = \{L, L, D, D, D, D\}$. Es ist zu beachten, dass M in diesem Fall keine Menge ist, da die Objekte nicht wohlunterschieden sind. Zur Angabe der Vielfachheit der Objekte kann die Multimenge M auch durch $M = \{L^{(2)}, D^{(4)}\}$ dargestellt werden. Unter einer festen Reihenfolge von Operationen der Form "LKW bedient Kunde" können sie als ununterscheidbare Objekte aufgefasst werden. Analog gilt dies für die Operationen bezüglich der Drohne. Die Anzahl aller Permutationen der Objekte einer solchen Multimenge $M = \{a_1^{(v_1)}, a_2^{(v_2)}, \dots, a_l^{(v_l)}\}$ mit $n := \sum_{k=1}^l v_k$ ist der *Multinomialkoeffizient*

$$\binom{n}{v_1, \dots, v_l} = \frac{n!}{v_1! \cdot \dots \cdot v_l!}$$

Dies kann zum Beispiel (Tittmann, 2019) (Seite 4) entnommen werden. Nun wird das obige Beispiel wieder aufgegriffen. Es gibt zwei Operationen der Form "LKW bedient Kunden" und 2 Kunden, die von der Drohne bedient werden, also 4 Operationen hinsichtlich der Drohne. Es gibt also

$$\binom{2+4}{2, 4} = \frac{6!}{2! \cdot 4!} = 15$$

Möglichkeiten, die Operationen für Beispiel 3.2.3.

Lemma 3.2.5. Die Anzahl aller partiell zulässigen Lösungen für n Kunden und die Menge T der Kunden, die nur durch den LKW bedient werden können ist

$$\sum_{k=0}^{n-|T|-1} \binom{n-|T|-1}{k} (n-1-k)! \binom{k+n-1}{2k} k!.$$

Beweis. Der Startkunde i^* wird nicht berücksichtigt, da dieser festgelegt ist. Der Laufindex k repräsentiert die Anzahl der Kunden, die durch die Drohne bedient werden. Die Anzahl der Kunden, bei denen das möglich ist, ist $n - |T| - 1$. Gesucht werden dann alle k -elementigen Teilmengen unter den Kunden, die durch die Drohne bedient werden können. Das sind $\binom{n-1-|T|}{k}$ viele. Wenn k Kunden

durch die Drohne bedient werden, gibt es $(n - 1 - k)!$ Reihenfolgefestlegungen bezüglich der Kunden, die durch den LKW bedient werden. Mit Anmerkung 3.2.4 ergibt sich für eine feste Reihenfolge von k Kunden, die durch die Drohne bedient werden und $n - 1 - k$ Kunden, die der LKW bedient die Anzahl aller Möglichkeiten, diese zu kombinieren:

$$\begin{aligned} \binom{(n-1-k)+2k}{(n-1-k), 2k} &= \frac{((n-1-k)+2k)!}{(n-1-k)! \cdot 2k!} = \frac{((n-1-k)+2k)!}{(n-1-k+2k-2k)! \cdot 2k!} \\ &= \binom{n-1-k+2k}{2k} = \binom{k+n-1}{2k} \end{aligned}$$

Dies ist die Anzahl aller partiell zulässigen Vektoren, für k Drohnen-Kunden und $n - 1 - k$ LKW-Kunden mit jeweils festen Reihenfolgen. Unter den Kunden, die durch die Drohne bedient werden, gibt es $k!$ Reihenfolgefestlegungen. \square

Beispiel 3.2.6. Sei $N = \{1, 2, 3, 4\}$ und $i^* \in N$ ein beliebiger aber fester Startkunde (z.B. $i^* = 2$). Außerdem sei $T = \{1\}$. Dann gibt es

$$\sum_{k=0}^2 \binom{2}{k} (3-k)! \binom{k+3}{2k} k! = 6 + 24 + 10 = 40$$

Möglichkeiten und die folgenden partiell zulässigen Lösungen:

k	Möglichkeiten		
0	(2, 10, 1, 3, 4)	(2, 10, 1, 4, 3)	
	(2, 10, 3, 1, 4)	(2, 10, 4, 1, 3)	
	(2, 10, 3, 4, 1)	(2, 10, 4, 3, 1)	
1	(2, 10, 1, 4, 7, 11)	(2, 10, 4, 7, 1, 11)	(2, 10, 1, 3, 8, 12)
	(2, 10, 4, 1, 7, 11)	(2, 10, 7, 4, 1, 11)	(2, 10, 3, 1, 8, 12)
	(2, 10, 1, 7, 4, 11)	(2, 10, 7, 4, 11, 1)	(2, 10, 1, 8, 3, 12)
	(2, 10, 7, 1, 4, 11)	(2, 10, 7, 11, 4, 1)	(2, 10, 8, 1, 3, 12)
	(2, 10, 1, 7, 11, 4)	(2, 10, 4, 7, 11, 1)	(2, 10, 1, 8, 12, 3)
	(2, 10, 7, 11, 1, 4)	(2, 10, 7, 1, 11, 4)	(2, 10, 8, 1, 12, 3)
	(2, 10, 8, 12, 1, 3)	(2, 10, 3, 8, 1, 12)	(2, 10, 8, 3, 1, 12)
	(2, 10, 8, 3, 12, 1)	(2, 10, 8, 12, 3, 1)	(2, 10, 3, 8, 12, 1)
2	(2, 10, 1, 7, 11, 8, 12)	(2, 10, 1, 8, 12, 7, 11)	
	(2, 10, 7, 1, 11, 8, 12)	(2, 10, 8, 1, 12, 7, 11)	
	(2, 10, 7, 11, 1, 8, 12)	(2, 10, 8, 12, 1, 7, 11)	
	(2, 10, 7, 11, 8, 1, 12)	(2, 10, 8, 12, 7, 1, 11)	
	(2, 10, 7, 11, 8, 12, 1)	(2, 10, 8, 12, 7, 11, 1)	

Der folgenden Tabelle lassen sich die Anzahl aller Möglichkeiten in Abhängigkeit der Anzahl der Kunden n und $|T|$ bis $n = 9$ und $|T| = 4$ entnehmen

	$ T = 0$	$ T = 1$	$ T = 2$	$ T = 3$	$ T = 4$
$n = 1$	1				
$n = 2$	1	1			
$n = 3$	10	5	2		
$n = 4$	78	40	18	6	
$n = 5$	816	426	204	84	24
$n = 6$	10680	5640	2796	1260	480
$n = 7$	167760	89280	45216	21384	9120
$n = 8$	3074400	1645200	845280	411840	186480
$n = 9$	64391040	34599600	17961120	8929440	4196160

3.3 Constraint Programming Ansatz

Das folgende Modell ist eines bezüglich des diskreten Falls. Begrifflichkeiten und Notation bezüglich CP, wie z.B. Intervallvariablen sind (Laborie, Rogerie, Shaw & Vilím, 2018) zu entnehmen. Es wird ein CP-Ansatz ähnlich zu (Tang et al., 2019) vorgestellt. Hierbei ist jedoch zu beachten, dass die Drohne bei jedem beliebigen Kunden gestartet und aufgelesen werden kann, auch wenn dieser Kunde bereits durch die Drohne oder den LKW bedient wurde.

```

int n=..;
range or=0..1;
range SepOrMeet=0..1;
range C1=1..n;
range C2=0..2*n;

int p[C1]=..;
int r[C1]=..;
int d[C1]=..;
int istar=..;
int vt=..;
int vd=..;
{int} T=..;
int Id[j in C1]=j;
tuple M { int c1; int c2; int value; };
{M} Dist ={{<i,j, ftoi(vt*(abs(p[i]-p[j]))))> | i in C1, j in C1};
dvar interval Serv[C1] optional;
dvar interval DorT[C1][or] optional;
dvar interval Tr[C1][C2] optional;
dvar interval Sep[i in C1][j in C2] optional
    end (0, d[i])
    size ftoi(vd*(abs(p[i]-p[j])));
dvar interval Meet[i in C1][j in C1] optional
    start (r[i], INTERVAL_MAX)
    size ftoi(vd*(abs(p[i]-p[j])));
dvar interval SepMeet[i in C1][j in SepOrMeet] optional;
dvar sequence seqDr in all(j in C1) DorT[j][1];
dvar sequence seqTr in all(j in C1, k in C2) Tr[j][k]
    types all(j in C1, k in C2) j;

```

```
dexpr int makespan = max(j in C1,k in C2)endOf(Tr[j][k]);
minimize makespan;
```

```
subject to {
forall (i in C1)
  presenceOf(Serv[i]);
  presenceOf(DorT[istar][0]);
  presenceOf(DorT[istar][1]);
  first(seqDr, DorT[istar][1]);
  first(seqTr, Tr[istar][0]);
  forall (i in T)
    presenceOf(DorT[i][0]);
  forall(i in C1:i!=istar)
    alternative(Serv[i], all(j in or) DorT[i][j]);
  forall(i in C1)
    presenceOf(DorT[i][1]) => presenceOf(SepMeet[i][0]);
    presenceOf(DorT[i][1]) => presenceOf(SepMeet[i][1]);
    presenceOf(DorT[i][0])=>presenceOf(Tr[i][0]);
    alternative(SepMeet[i][0], all(j in C1) Meet[i][j]);
    alternative(SepMeet[i][1], all(j in C1) Sep[i][j]);
    span(DorT[i][1], all(j in SepOrMeet) SepMeet[i][j]);
    endBeforeStart(SepMeet[i][0],SepMeet[i][1]);

forall(i in C1)
  forall(j in C1)
    presenceOf(Sep[i][j]) => presenceOf(Tr[j][i]) ;
    presenceOf(Meet[i][j]) =>presenceOf(Tr[j][i+n]);
    startBeforeStart(Tr[i][j],DorT[j][1]);
    startBeforeEnd(DorT[j][1],Tr[i][j]);
    endBeforeStart(DorT[j][1],Tr[i][j+n]);
noOverlap(seqTr,Dist, true);
noOverlap(seqDr); }
```

Die Intervallvariable $Serv[i]$ wird genutzt, damit der Kunde i bedient werden kann. Diese Variable muss für jeden Kunden verfügbar sein. Ihr wird entweder die Intervallvariable $DorT[i][0]$ (Kunde i wird durch den LKW bedient) oder $DorT[i][1]$ (Kunde i wird durch die Drohne bedient) zugeordnet. Wird ein Kunde durch die Drohne bedient, so werden die Intervallvariablen $SepMeet[i][0]$ (Flugzeit der Drohne vom Trennungsort zu Kunde i) und $SepMeet[i][1]$ (Flugzeit der Drohne von Kunde i zum Begegnungsort) zugewiesen. Für $SepMeet[i][1]$ wird genau eine Intervallvariable unter den Intervallvariablen $Meet[i][j]$ mit $j \in \{1, \dots, n\}$ genutzt, da diese den Begegnungsort bei Kunde j beschreibt. Die Größe von $Meet[i][j]$ entspricht der Flugzeit der Drohne von Kunde i zu Kunde j . Der Beginn dieser Intervallvariablen darf nicht vor dem Zeitpunkt r_i liegen, da andernfalls dieser Kunde nicht bedient werden würde. Analog gelten die Zusammenhänge der entsprechenden Intervallvariablen für die Intervallvariable $SepMeet[i][0]$. Die Drohnenroute setzt sich aus den Intervallvariablen $DorT[i][1]$ zusammen, die sich nicht überschneiden dürfen. Die LKW-Route setzt sich aus den Intervallvariablen $Tr[i][j]$ zusammen, wobei i der Kunde ist, der durch den LKW bedient wird, falls $j = 0$ ist.

Ansonsten findet entweder der Start oder das Aufladen der Drohne am Ort p_i statt, ohne dass der Kunde i bedient wird. Dabei ist $j \in \{1, \dots, n\}$, wenn der LKW die Drohne bei Kunde i startet, damit diese Kunde j bedienen kann. Der Fall $j \in \{n+1, \dots, 2n\}$ liegt vor, wenn der LKW der Drohne bei Kunde $j-n$ begegnet, nachdem diese Kunde i bedient hat. Diese Intervallvariablen weisen eine Größe von 0 auf. Es werden die Reisezeiten mithilfe der Reisezeitmatrix `Dist` berücksichtigt (entsprechend im Scheduling zu Setupzeiten). Auch in dieser Route dürfen sich die Intervallvariablen nicht überschneiden. Das folgende Beispiel zeigt den Zusammenhang der Intervallvariablen mit einer zugehörigen Lösung.

Beispiel 3.3.1. Seien $n = 7$ Kunden gegeben. Der Startkunde ist Kunde $i^* = 3$. Abbildung 3.4 zeigt eine Lösung, bei welcher der LKW von Kunde 3 zu Kunde 4 fährt und diesen auch bedient. Die Intervallvariable `Tr[4][0]` hat eine Länge von 0. Hier ist zu beachten, dass zwischen den Intervallvariablen der Sequenz `SeqTr` Pufferzeiten berücksichtigt werden, die durch die Matrix `Dist` festgelegt sind und die Reisezeiten des LKWs zwischen den Kunden darstellen. Anschließend wird auch bei Kunde 4 die Drohne gestartet, damit diese Kunde 6 bedient. Dies drückt die Intervallvariable `Tr[4][6]` aus. Während der LKW bei Kunde 4 ist, muss die Drohne gestartet werden. Dies wird mit den Restriktionen `startBeforeStart(Tr[4][6], DorT[6][1])` und `startBeforeEnd(DorT[6][1], Tr[4][6])` beschrieben. Der Start der Drohne muss also innerhalb der Zeit stattfinden, während der LKW bei Kunde 4 ist. Nun wird `DorT[6][1]` durchgeführt. Diese Intervallvariable beinhaltet die Intervallvariablen `SepMeet[6][1]` und `SepMeet[6][0]`, wegen der Bedingung `span(DorT[6][1], all(j in SepOrMeet) SepMeet[6][j])`. Die Intervallvariable `DorT[6][1]` beinhaltet eine der Intervallvariablen `Sep[6][i]`, deren Länge gleich der Reisezeit der Drohne von i nach 6 ist. In diesem Beispiel ist es die Variable `Sep[6][4]`. Wegen der Bedingung `presenceOf(Sep[6][4])=>Tr[4][6]`, ist auch die Intervallvariable `Tr[4][6]` notwendig. Darüber hinaus enthält die Intervallvariable `DorT[6][1]` eine der Intervallvariablen `Meet[6][j]`, deren Länge gleich der Reisezeit der Drohne von 6 nach j ist. In der vorliegenden Lösung ist dies die Variable `Meet[6][5]`, daher muss auch die Intervallvariable `Tr[5][6+7]` vorhanden sein. Die Intervallvariablen `SepMeet[6][1]` und `SepMeet[6][0]` werden hintereinander ausgeführt. Dies wird durch die Restriktion `endBeforeStart(SepMeet[6][0], SepMeet[6][1])` erzwungen. Während sich die Drohne mit Kunde 6 befasst, bedient der LKW Kunde 2, da die Intervallvariable `Tr[2][0]` der Sequenzvariable `TrSeq` zwischen den Intervallvariablen `Tr[4][6]` und `Tr[5][13]` liegt. Insgesamt ergeben sich die Sequenzvariable `TrSeq` mit der Reihenfolge `(Tr[4][0], Tr[4][6], Tr[2][0], Tr[5][0], Tr[5][13], Tr[5][7], Tr[1][0], Tr[3][0], Tr[3][14])` und die Sequenzvariable `DrSeq` mit der Reihenfolge `(DorT[6][1], DorT[7][1])`.

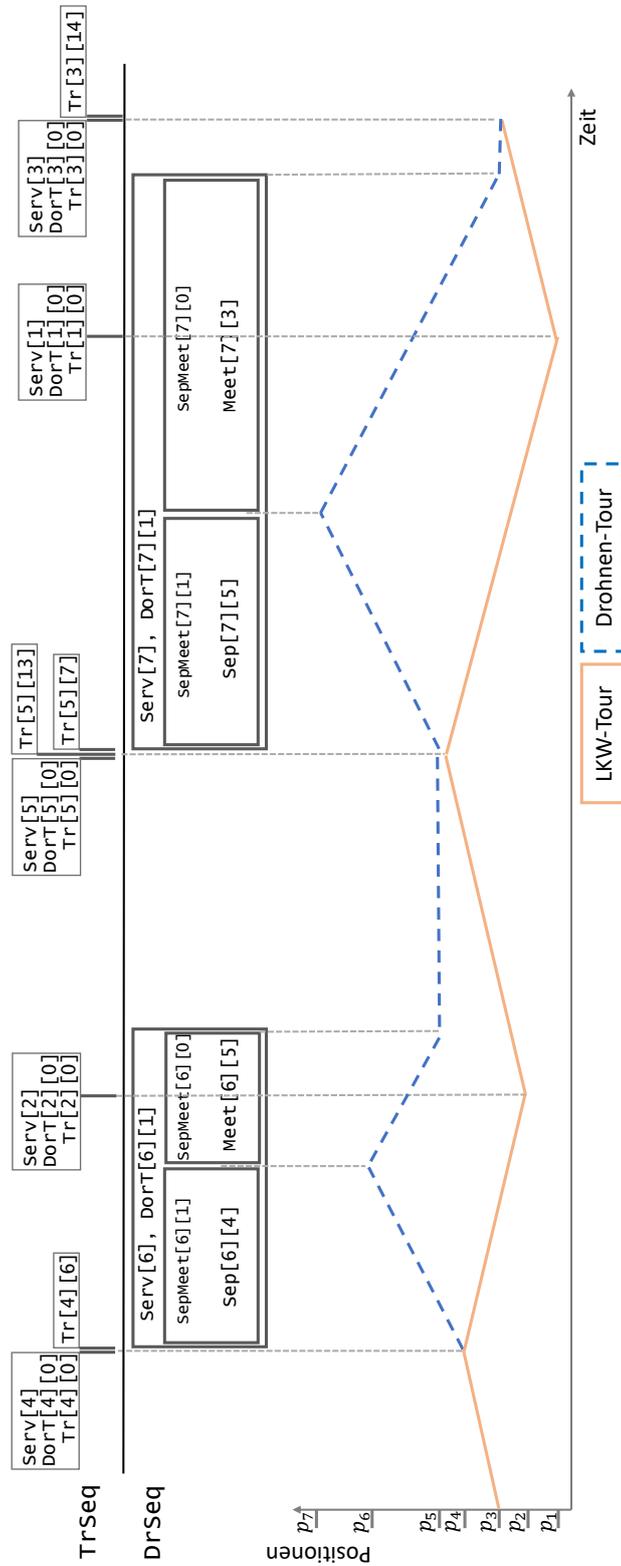


Abbildung 3.4: Lösung einer diskreten Line-TSPwD Instanz mit zugehörigen Intervallvariablen

Kapitel 4

Komplexität verschiedener Varianten

4.1 Grundlegende Begriffe der Komplexitätstheorie

In diesem Abschnitt werden grundlegende Begriffe der Komplexitätstheorie definiert. In der Literatur bieten unter anderem (Wegener, 2005) oder (Garey & Johnson, 1979) eine Vertiefung der Theorie. Das Konzept der \mathcal{NP} -Schwere beruht auf Entscheidungsproblemen. Dies sind Probleme, die für eine gegebene Instanz mit einer Entscheidungsfrage verbunden sind, die mit *Ja* oder *Nein* zu beantworten ist. Ein Beispiel hierfür ist folgendes Problem, welches in der Literatur als Rucksackproblem (engl. knapsack problem) bezeichnet wird (siehe z.B. (Garey & Johnson, 1979)). Hierbei sind n Elemente $1, \dots, n$ gegeben, welchen jeweils der Nutzen $p_i \in \mathbb{Z}_{>0}$ und das Gewicht $w_i \in \mathbb{Z}_{>0}$ für $i = 1, \dots, n$ zugewiesen ist, sowie zwei weitere Elemente $B, K \in \mathbb{Z}_{>0}$. Die Entscheidungsfrage lautet „Existiert eine Teilmenge A von $\{1, \dots, n\}$, sodass die Summe aller Nutzwerte p_i für $i \in A$ gleich dem vorgegebenen Gesamtnutzen K ist, unter der Bedingung, dass die Summe aller Gewichte w_i für $i \in A$ ein vorgegebenes Gesamtgewicht B nicht überschreitet?“

Im Folgenden wird das Konzept einer sogenannten Turingmaschine vorgestellt, welches (Wegener, 2005) entnommen ist. Ein Zweck dieses theoretischen Konstrukts ist die Formalisierung der Begriffe bezüglich der Komplexitätstheorie, die nun beruhend auf diesem Modell definiert werden können und ebenfalls (Wegener, 2005) entnommen sind.

Definition 4.1.1. Eine *deterministische Turingmaschine (DTM)* ist ein 7-Tupel $(Q, \Sigma, \Gamma, B, q_0, \delta, F)$ mit

- Q ist eine endliche Menge, die sogenannte *Zustandsmenge*
- Γ ist eine endliche Menge, das sogenannte *Bandalphabet*
- $\Sigma \subset \Gamma$, das sogenannte *Eingabealphabet*
- $B \in \Gamma \setminus \Sigma$, das sogenannte *Leerzeichen* (engl. blank)

- $q_0 \in Q$, der sogenannte *Anfangszustand*
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$, die sogenannte *Überföhrungsfunktion*. Die Menge $\{L, R, N\}$ ist die *Richtungsmenge*.
- $F \subset Q$, die sogenannte Menge der *akzeptierenden Zustände*.

Dabei ist die Vorgehensweise wie folgt zu verstehen. Es liegt ein zweiseitiges unbeschränktes Band mit Zellen vor, also die Zellen $\dots - 3, -2, -1, 0, 1, 2, 3, \dots$, die jeweils nur ein Element aus Γ enthalten können. Der eindeutige Rechnerkopf (dieser kann lesen und schreiben) befindet sich immer genau bei einer Zelle. Die Maschine befindet sich immer in einem Zustand q und führt gemäß der Überföhrungsfunktion δ (das Programm) Befehle aus. Dies geschieht wie folgt: Sei die Maschine in Zustand q und sei $a \in \Gamma$ der Zelleintrag, den der Rechnerkopf einliest. Dann liefert $\delta(q, a) = (q', a', d)$ den neuen Zustand q' , den neuen Zelleninhalt a' der gelesenen Zelle und die neue Position des Rechnerkopfs. Ist $d = R$, so befindet er sich direkt rechts von der vorherigen Zelle, ist $d = L$, so befindet er sich direkt links von der vorherigen Zelle, ist $d = N$, so hat der Rechnerkopf seine Position nicht verändert.

Das Programm soll aber nun in Abhängigkeit einer Eingabe $(w_1, \dots, w_n) \in \Sigma^* = \{x \mid x \text{ ist eine endliche Folge von Elementen aus } \Sigma\}$ ablaufen. Hierbei wird w_i in die Zelle i geschrieben, und alle übrigen Zellen erhalten den Eintrag B (Leerzeichen). Die Maschine beginnt mit dem Anfangszustand q_0 .

Definition 4.1.2. Die *Komplexitätsklasse* \mathcal{P} besteht aus allen Problemen, für die eine deterministische Turingmaschine existiert, deren worst case Rechenzeit (maximale Anzahl an Rechenschritten bis in einen Endzustand) polynomiell in der Eingabelänge beschränkt ist.

Definition 4.1.3. Wird die Überföhrungsfunktion einer deterministischen Turingmaschine ersetzt durch eine *Überföhrungsrelation*

$$\delta = (Q \times \Gamma) \times (Q \times \Gamma \times \{L, R, N\}),$$

so spricht man von einer *nichtdeterministische Turingmaschine*.

Das bedeutet, dass von einem Zustand aus mehrere mögliche Rechenschritte existieren. Sei $(q, a) \in (Q, \Gamma)$ gegeben, dann sind alle $(q', a', d) \in (Q \times \Gamma \times \{L, R, N\})$ mit $(q, a, q', a', d) \in \delta$ mögliche Rechenschritte. Im Gegensatz dazu gibt es bei der deterministischen Turingmaschine einen eindeutigen Rechenschritt, welcher durch die Überföhrungsfunktion bestimmt wird.

Definition 4.1.4. Sei eine nichtdeterministische Turingmaschine gegeben. Eine Eingabe $w \in \Sigma$ wird *akzeptiert*, falls ein Rechenweg existiert, der zu einen akzeptierenden Zustand führt. Die *Rechenzeit* für eine akzeptierte Eingabe ist die Anzahl der Rechenschritte für einen kürzesten Rechenweg. Die *worst case Rechenzeit* bezüglich der Eingabelänge n ist das Maximum der Rechenzeit unter allen Eingaben der Länge n .

Definition 4.1.5. Die *Komplexitätsklasse* \mathcal{NP} besteht aus allen Entscheidungsproblemen, für die eine nichtdeterministische Turingmaschine existiert, deren worst case Rechenzeit polynomiell beschränkt ist.

Werden also alle möglichen Lösungen gleichzeitig getestet, und ist das Überprüfen einer Lösung polynomiell beschränkt, so liegt das Problem in \mathcal{NP} . Probleme die in \mathcal{P} liegen, liegen also somit auch in \mathcal{NP} .

Definition 4.1.6. Eine *Reduktion* r eines Entscheidungsproblems L_1 auf ein Entscheidungsproblem L_2 ist eine Funktion, die jede Instanz von L_1 in eine Instanz von L_2 transformiert, sodass genau dann für alle Instanzen I von L_1 , die mit *Ja* beantwortet werden, auch $r(I)$ in L_2 mit *Ja* beantwortet werden.

Definition 4.1.7. Ein Entscheidungsproblem L heißt \mathcal{NP} -*schwer*, wenn für alle Entscheidungsprobleme $K \in \mathcal{NP}$ jeweils eine polynomielle Reduktion auf L existiert.

Definition 4.1.8. Ein Entscheidungsproblem heißt \mathcal{NP} -*vollständig*, wenn es in \mathcal{NP} liegt und \mathcal{NP} -schwer ist.

Das bedeutet, dass ein Problem als \mathcal{NP} -schwer bezeichnet wird, wenn es mindestens so *schwer* ist, wie die *schwierigsten* Probleme in \mathcal{NP} .

Es sei hier angemerkt, dass sich \mathcal{NP} -schwere Probleme einteilen lassen können in die Menge der \mathcal{NP} -schweren Problem im *schwachen Sinne* und in die Menge der \mathcal{NP} -schweren Probleme im *strengen Sinne*.

Definition 4.1.9. Sei ein Entscheidungsproblem L gegeben. Dann bezeichnet $l_L(I)$ die Eingabelänge für eine Instanz I und $\max_L(I)$ den Wert des Eingabeparameters mit maximalen Wert.

Definition 4.1.10. Ein *pseudopolynomieller Algorithmus* ist ein Algorithmus, dessen Laufzeit in $\mathcal{O}(p(\max_L(I), l_L(I)))$ für ein Polynom p liegt.

Definition 4.1.11. Ein Problem, das \mathcal{NP} -schwer ist und für welches ein pseudopolynomieller Algorithmus existiert, der dieses Problem löst, heißt \mathcal{NP} -schwer im *schwachen Sinne*.

Ein Beispiel hierfür ist das Partitions Problem. Dabei sind ganzzahlige Werte $S(i)$ für $i = 1, \dots, n$ gegeben mit $\sum_{i \in \{1, \dots, n\}} s(i) = B$ für vorgegebenes B . Es wird nach einer Teilmenge $A \subset \{1, \dots, n\}$ gesucht, sodass $\sum_{i \in A} s(i) = B/2$ ist. Dieses ist \mathcal{NP} -vollständig (siehe (Garey & Johnson, 1979), S. 47), es gibt aber einen Algorithmus, der dieses Problem in $\mathcal{O}(nB)$ löst (siehe (Garey & Johnson, 1979) S. 90f). Da B aber ein Inputparameter ist und nicht notwendigerweise polynomiell in der Eingabelänge beschränkt sein muss, ist der Algorithmus nicht polynomiell, jedoch pseudopolynomiell.

Definition 4.1.12. Seien ein Entscheidungsproblem L und ein Polynom p gegeben. Mit L_p wird die Gesamtheit aller Instanzen des Entscheidungsproblems bezeichnet, wobei der Wert jedes Inputparameters durch $p(l)$ beschränkt wird. Hierbei ist l die Eingabelänge.

Definition 4.1.13. Ein Entscheidungsproblem L heißt \mathcal{NP} -*schwer im strengen Sinne*, wenn ein Polynom existiert, sodass L_p \mathcal{NP} -schwer ist.

Sind also die Parameter polynomiell beschränkt (durch ein Polynom in der Eingabelänge) und ist das Problem \mathcal{NP} -schwer für solch eine Beschränkung, so ist das Problem \mathcal{NP} -schwer im strengen Sinne. Beispiele hierfür sind das Satisfiability (SAT) Problem, welches kein sogenanntes Zahlenproblem ist, aber auch das TSP.

Definition 4.1.14. ((Garey & Johnson, 1979), S. 101) Seien zwei Entscheidungsprobleme L_1 und L_2 gegeben, wobei I_{L_1} und I_{L_2} die Mengen aller Instanzen des zugehörigen Entscheidungsproblems sind. Eine Abbildung $f : I_{L_1} \rightarrow I_{L_2}$ heißt *pseudo-polynomielle Transformation*, wenn

1. Die Antwort auf $I \in I_{L_1}$ ist genau dann *Ja*, wenn die Antwort auf $f(I)$ *Ja* ist.
2. f lässt sich zeitlich polynomiell in den Variablen $\max_{L_1}(I)$ und $l_{L_1}(I)$ berechnen.
3. Es gibt ein Polynom q_1 , sodass $\forall I \in I_{L_1} : q_1(l_{L_2}(f(I))) \geq l_{L_1}(I)$ ist.
4. Es gibt ein Polynom q_2 in zwei Variablen, sodass $\max_{L_2}(f(I)) \leq q_2(\max_{L_1}(I), l_{L_1}(I))$ ist.

Lemma 4.1.15. ((Garey & Johnson, 1979), S. 101) Sei das Entscheidungsproblem L_1 \mathcal{NP} -vollständig im strengen Sinne. Gibt es eine pseudo-polynomielle Transformation von L_1 nach $L_2 \in \mathcal{NP}$, so ist L_2 \mathcal{NP} -vollständig im strengen Sinne.

In (Aaronson, 2016) wird der aktuelle Stand der Forschung zur Frage der Gleichheit bzw. Ungleichheit von \mathcal{P} und \mathcal{NP} behandelt, welche weiterhin unbeantwortet ist. In einem Großteil der wissenschaftlichen Literatur, die sich mit Komplexitätstheoretischen Fragestellungen beschäftigt, wird $\mathcal{P} \neq \mathcal{NP}$ angenommen.

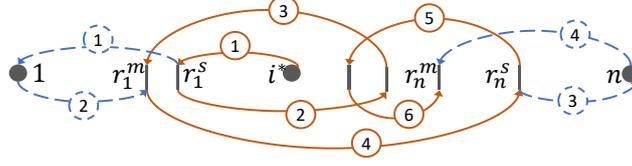
Um also nun zu zeigen, dass ein Problem tatsächlich \mathcal{NP} -schwer ist, muss eine Reduktion auf ein \mathcal{NP} -vollständiges Problem gefunden werden. Hier ist das SAT-Problem (Satisfiability) der Grundstein, denn das Theorem von Cook (Cook, 1971) besagt, dass dieses \mathcal{NP} -vollständig ist. Im Beweis wird kein bestimmtes Problem für eine Reduktion herangezogen, da zu diesem Zeitpunkt noch kein \mathcal{NP} -vollständiges Problem bekannt war. Stattdessen wurde für ein beliebiges Problem in \mathcal{NP} eine Reduktion konstruiert. Von nun an befasst sich die vorliegende Arbeit mit der Komplexität verschiedener Szenarien des Line-TSPwD. Der erste Fall, der nun betrachtet wird, setzt $v^D \leq v^T$ voraus. Es wird gezeigt, dass die Drohne in diesem Fall maximal einen Kunden bedient.

4.2 Keine Zeitfenster und $v^D \leq v^T$

Definition 4.2.1. Sei eine Kundenmenge $C = \{1, \dots, n\}$ gegeben. Für $j \in C$ ist $L_j := \{i \in C \mid i < j\}$ und $R_j := \{i \in C \mid i > j\}$.

Lemma 4.2.2. Sei eine Line-TSPwD Instanz ohne Zeitfenster gegeben mit Kundenmenge $C = \{1, \dots, n\}$. Dann gibt es $i, j \in C$ mit $i \leq i^* \leq j$, sodass eine optimale Lösung existiert, in welcher genau die Kunden in L_i und R_j durch die Drohne bedient und die aus $C \setminus (L_i \cup R_j)$ durch den LKW bedient werden.

Beweis. Unter der Annahme, dass eine optimale Lösung existiert, bei der ein Kunde $i < i^*$ durch die Drohne und ein Kunde $j < i$ durch den LKW bedient wird, kann der LKW den Kunden i bedienen, ohne dass sich der Zielfunktionswert verschlechtert, sobald er an diesem vorbeifährt. Daher existiert eine optimale Lösung, bei der die Drohne links vom Startkunden i^* keinen Kunden bedient, der sich rechts neben einem Kunden befindet, welcher durch den LKW

Abbildung 4.1: Drohne bedient Kunde 1 und Kunde n

bedient wird. Analog gilt dies auch für die rechte Seite von i^* . Somit existiert eine optimale Lösung, bei der die Drohne genau die Kunden der Kundenmenge $L_j \cup R_k$ mit $j < i^* < k$ bedient. \square

Lemma 4.2.3. *Sei ein Line-TSPwD mit $v_D \leq v_T$ und n Kunden gegeben. Dann existiert ein optimale Lösung, bei der die Drohne entweder ausschließlich Kunden 1, Kunden n , oder keinen Kunden besucht. Dies gilt sowohl für die Minimierung von $\max\{t_i\}$, als auch für die Minimierung von $\max\{t_i, t_i^m\}$.*

Beweis. In 4.2.2 wurde gezeigt, dass es eine optimale Lösung gibt, in welcher der LKW genau die Kunden der Menge $S = \{i \in \{1, \dots, n\} : l \leq i \leq k\}$ für $l, k \in \{1, \dots, n\}$ bedient und die Drohne die übrigen Kunden der Menge $\{1, \dots, n\} \setminus S$. Im Folgenden wird gezeigt, dass eine optimale Lösung existiert, bei der die Drohne nicht Kunde 1 und Kunde n bedient. Angenommen es gibt eine optimale Lösung, bei der die Drohne sowohl Kunde 1, als auch Kunde n bedient. O.B.d.A. wird Kunde 1 vor Kunde n bedient, wie in Abbildung 4.1. Dann gelten die folgenden Abschätzungen:

$$\begin{aligned} t_1^s &\geq \frac{|r_1^s - p_{i^*}|}{v^T} & t_1 &\geq t_1^s + \frac{|p_1 - r_1^s|}{v^D} & t_1^m &\geq t_1 + \frac{|p_1 - r_1^m|}{v^D} \\ t_n^s &\geq t_1^m + \frac{|r_1^m - r_n^s|}{v^T} & t_n &\geq t_n^s + \frac{|r_n^s - p_n|}{v^D} & t_n^m &\geq t_n + \frac{|p_n - r_n^m|}{v^D} \end{aligned}$$

Somit ergeben sich folgenden Abschätzungen:

$$\begin{aligned} t_n &\geq t_n^s + \frac{|r_n^s - p_n|}{v^D} \geq t_1^m + \frac{|r_1^m - r_n^s|}{v^T} + \frac{|r_n^s - p_n|}{v^D} \\ &\geq t_1 + \frac{|p_1 - r_1^m|}{v^D} + \frac{|r_1^m - r_n^s|}{v^T} + \frac{|r_n^s - p_n|}{v^D} \\ &\geq t_1^s + \frac{|p_1 - r_1^s|}{v^D} + \frac{|p_1 - r_1^m|}{v^D} + \frac{|r_1^m - r_n^s|}{v^T} + \frac{|r_n^s - p_n|}{v^D} \\ &\geq \frac{|r_1^s - p_{i^*}|}{v^T} + \frac{|p_1 - r_1^s|}{v^D} + \frac{|p_1 - r_1^m|}{v^D} + \frac{|r_1^m - r_n^s|}{v^T} + \frac{|r_n^s - p_n|}{v^D} \\ &\geq \frac{1}{v^T} (|r_1^s - p_{i^*}| + |p_1 - r_1^s| + |p_1 - r_1^m| + |r_1^m - r_n^s| + |r_n^s - p_n|) \\ &\geq \frac{1}{v^T} (|r_1^s - p_{i^*}| + |p_1 - r_1^s| + |p_1 - r_1^m| + |r_1^m - r_n^s| + |r_n^s - p_n|) \\ &= \frac{|p_1 - p_{i^*}| + |p_1 - p_n|}{v^T} \end{aligned}$$

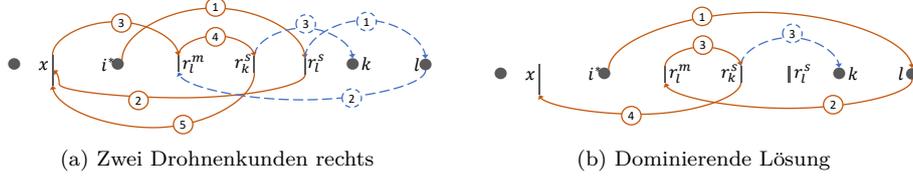


Abbildung 4.2: Lösung, bei der die Drohne zwei Kunden rechts von i^* bedient in (a) und eine dominierende Lösung in (b).

Somit kann festgehalten werden, dass eine Lösung, bei der die Drohne Kunde 1 und Kunde n bedient, von einer Lösung dominiert wird, bei der der LKW den Kunden 1 und den Kunden n bedient. Dies gilt sowohl für die zu minimierende Zielfunktion $\max\{t_i\}$, als auch für die zu minimierende Zielfunktion $\max\{t_i^m, t_i\}$.

Es wird nun gezeigt, dass maximal ein Kunde durch die Drohne bedient wird. Angenommen, es gibt eine optimale Lösung, bei der die Drohne zwei Kunden l und k bedient, die sich entweder links von i^* oder rechts von i^* befinden, und Kunde l vor Kunde k bedient wird (siehe Abbildung 4.2 (a)). Es sei r_l^s der Trennungsort und t_l^s der Trennungzeitpunkt. In diesem Fall wendet der LKW umgehend, da andernfalls ein Start der Drohne näher am Kunden l möglich gewesen wäre. Anschließend wendet der LKW am Ort x zur Zeit t^{x_1} . Somit ist

$$t^{x_1} \geq t_l^s + \frac{|x - r_l^s|}{v^T}.$$

Der früheste Zeitpunkt, zu dem sich die Drohne am Ort r_l^m befindet, ist

$$t_l^m \geq t_l^s + \frac{|r_l^s - p_l| + |p_l - r_l^m|}{v^D}.$$

Im Anschluss erfolgt eine zweite Trennung von LKW und Drohne am Ort r_k^m , der nicht weiter von p_k entfernt ist als r_l^m . Zum Zeitpunkt t_k^s wendet der LKW erneut, da er sonst die Drohne näher am Kunden hätte starten können. Somit ist

$$t_k^s \geq t_l^m + \frac{|r_l^m - r_k^s|}{v^T} \quad t_k \geq t_k^s + \frac{|r_k^s - p_k|}{v^D}$$

Es sei t^{x_2} der Zeitpunkt, zu dem der LKW nach dem Start der Drohne, die nun Kunde k bedient, wieder am Ort x vorbeifährt. Dann ergeben sich die folgenden Abschätzungen:

$$\begin{aligned} t^{x_2} &\geq t_k^s + \frac{|x - r_k^s|}{v^T} \geq t_l^m + \frac{|r_l^m - r_k^s|}{v^T} + \frac{|x - r_k^s|}{v^T} \\ &\geq t_l^s + \frac{|r_l^s - p_l| + |p_l - r_l^m|}{v^D} + \frac{|r_l^m - r_k^s|}{v^T} + \frac{|x - r_k^s|}{v^T} \\ &\geq t_l^s + \frac{1}{v^T} (|r_l^s - p_l| + |p_l - r_l^m| + |r_l^m - r_k^s| + |x - r_k^s|) \\ &\geq t_l^s + \frac{1}{v^T} (|r_l^s - p_l| + |p_l - r_l^m| + r_l^m - r_k^s + |x - r_k^s|) \end{aligned}$$

$$= t_l^s + \frac{1}{v^T} (|r_l^s - p_l| + |p_l - r_k^s| + |x - r_k^s|)$$

Die LKW-Tour $r_l^s \rightarrow p_l \rightarrow r_k^s \rightarrow x$ endet nicht später bei x als in der vorherigen Lösung, und der LKW hat den Kunden l bedient. Wird die Drohne bei dieser Tour auch bei r_k^s gestartet, so erfolgt dies nicht zu einem späteren Zeitpunkt als in der vorherigen Lösung. Die folgenden Abschätzungen belegen dies:

$$\begin{aligned} t_k^s &\geq t_l^m + \frac{|r_l^m - r_k^s|}{v^T} \geq t_l^s + \frac{|r_l^s - p_l| + |p_l - r_l^m|}{v^D} + \frac{|r_l^m - r_k^s|}{v^T} \\ &\geq t_l^s + \frac{|r_l^s - p_l| + |p_l - r_l^m| + |r_l^m - r_k^s|}{v^T} \end{aligned}$$

Somit lässt sich eine dominierende Lösung wie in Abbildung 4.2 (b) ableiten. Dies gilt wieder für beide der betrachteten Zielfunktionen. Es existiert demnach eine optimale Lösung, bei der Kunde 1, Kunde n oder kein Kunde von der Drohne bedient wird. \square

Lemma 4.2.4. *Sei eine Line-TSPwD Instanz mit $v_D \leq v_T$ und n Kunden mit $1 < i^* < n$ gegeben. Dann ist*

$$\min \left\{ \begin{array}{l} \max_{i \in \{1, \dots, n\}} \{t_i\} = \\ \min_{x \in [p_{n-1}, p_n]} \left\{ \max \left\{ \begin{array}{l} \frac{1}{v^T} \cdot (2 \cdot x - p_{i^*} - p_1), \\ \frac{1}{v^D} \cdot (p_n - x) + \frac{1}{v^T} \cdot (x - p_{i^*}) \end{array} \right\} \right\}, \\ \min_{x \in [p_{i^*}, p_{n-1}]} \left\{ \max \left\{ \begin{array}{l} \frac{1}{v^T} \cdot (2 \cdot x - p_{i^*} - 2 \cdot p_1 + p_{n-1}), \\ \frac{1}{v^D} \cdot (p_n - x) + \frac{1}{v^T} \cdot (x - p_{i^*}) \end{array} \right\} \right\}, \\ \min_{x \in [p_1, p_2]} \left\{ \max \left\{ \begin{array}{l} \frac{1}{v^T} \cdot (-2 \cdot x + p_{i^*} + p_n), \\ \frac{1}{v^D} \cdot (x - p_1) + \frac{1}{v^T} \cdot (p_{i^*} - x) \end{array} \right\} \right\}, \\ \min_{x \in [p_2, p_{i^*}]} \left\{ \max \left\{ \begin{array}{l} \frac{1}{v^T} \cdot (-2 \cdot x + p_{i^*} + 2 \cdot p_n - p_2), \\ \frac{1}{v^D} \cdot (x - p_1) + \frac{1}{v^T} \cdot (p_{i^*} - x) \end{array} \right\} \right\}, \\ \frac{1}{v^T} \cdot (2 \cdot p_n - p_{i^*} - p_1), \\ \frac{1}{v^T} \cdot (-2 \cdot p_1 + p_{i^*} + p_n) \end{array} \right\}$$

Beweis. Wegen 4.2.3 gibt es eine optimale Lösung, in der entweder Kunde 1, Kunde n oder gar kein Kunde durch die Drohne bedient wird. In diesem Fall würde die Drohne maximal einmal gestartet. Bedient die Drohne Kunde n , so sind ausschließlich die Trennungsorte $x \in [p_{n-1}, p_n]$ oder $x \in [p_{i^*}, p_{n-1}]$ relevant. Wird hingegen Kunde 1 durch die Drohne bedient, so kommen ausschließlich die Trennungsorte $x \in [p_1, p_2]$ und $x \in [p_2, p_{i^*}]$ in Frage. Sei zunächst Kunde n derjenige, der durch die Drohne bedient wird, und sei $x \in [p_{n-1}, p_n]$ der Trennungsort. Dann fährt der LKW zunächst direkt zum Trennungsort und anschließend zu Kunde 1. Damit fährt der LKW zweimal die Strecke zwischen

x und p_{i^*} und einmal die Strecke zwischen p_{i^*} und p_1 . Damit wäre der LKW nach einer Zeit von $\frac{1}{v^T} \cdot (2 \cdot (x - p_{i^*}) + (p_{i^*} - p_1)) = \frac{1}{v^T} \cdot (2 \cdot x - p_{i^*} - p_1)$ fertig. Die Drohne erreicht Kunde n zum Zeitpunkt $t_n = \frac{1}{v^D} \cdot (p_n - x) + \frac{1}{v^T} \cdot (x - p_{i^*})$. Liegt der Trennungsort zwischen p_{i^*} und p_{n-1} , so fährt der LKW wieder direkt zum Trennungsort und anschließend zu Kunde 1. Da Kunde $n-1$ aber noch nicht bedient wurde, muss der LKW wieder zurückfahren bis zum Kunden $n-1$. Daher gilt $t_{n-1} = \frac{1}{v^T} \cdot (3 \cdot (x - p_{i^*}) + 2 \cdot (p_{i^*} - p_1) + p_{n-1} - x) = \frac{1}{v^T} \cdot (2 \cdot x - p_{i^*} - 2 \cdot p_1 + p_{n-1})$. Analoges gilt für den Fall, falls die Drohne Kunde 1 bedient.

Bedient die Drohne keinen Kunden, so fährt der LKW von p_{i^*} direkt nach p_1 und von hier nach p_n , oder der LKW von p_{i^*} direkt nach p_n und von hier nach p_1 . \square

Anmerkung 4.2.5. Seien zwei Abbildungen $f : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto mx + b$ und $g : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto nx + c$ mit $m, n, b, c \in \mathbb{R}$ und $m \neq n$ gegeben. Außerdem sei die Abbildung s gegeben durch:

$$s(f, g, [a, b]) = \begin{cases} z & \text{falls } f(z) = g(z) \text{ und } z \in [a, b] \\ \max\{f(a), g(a)\} & \text{sonst} \end{cases}$$

Dann ist

$$\min_{x \in [a, b]} \max\{f(x), g(x)\} = \min\{\max\{f(a), g(a)\}, \max\{f(b), g(b)\}, s(f, g, [a, b])\}$$

Damit ist ein Optimum für das Problem mit $v^D \leq v^T$ und ohne Zeitfenster in konstanter Zeit lösbar.

Beispiel 4.2.6. Betrachtet werden zwei Instanzen mit 5 Kunden. Eine der Instanzen führt zu einer optimalen Lösung gemäß des ersten Eintrags aus Lemma 4.2.4, und die andere Instanz führt zu einer optimalen Lösung gemäß des zweiten Eintrags. In Instanz 1 liegt die Geschwindigkeit der Drohne bei $v^D = 2$ und die des LKWs bei $v^T = 2, 5$. Der Startkunde ist $i^* = 3$. In Instanz 2 ist der Startkunde ebenfalls $i^* = 3$, jedoch $v^D = 1$ und $v^T = 2$. Die weiteren Parameter sowie eine optimale Lösung sind den beiden Tabellen aus 4.1 zu entnehmen. Die Lösungen sind in 4.3 visualisiert.

i	1	2	3	4	5	i	1	2	3	4	5
p_i	0	4	18	40	75	p_i	0	4	18	45	75
t_i	28.8	27.2	0.0	8.8	28.8	t_i	10.1	11.7	0.0	28.1	28.1
r_i^m	-	-	-	-	18	r_i^m	-	-	-	-	75
t_i^m	-	-	-	-	63	t_i^m	-	-	-	-	40.1
r_i^s	-	-	-	-	45	r_i^s	-	-	-	-	21.7
t_i^s	-	-	-	-	10.8	t_i^s	-	-	-	-	1.5

(a) Instanz 1

(b) Instanz 2

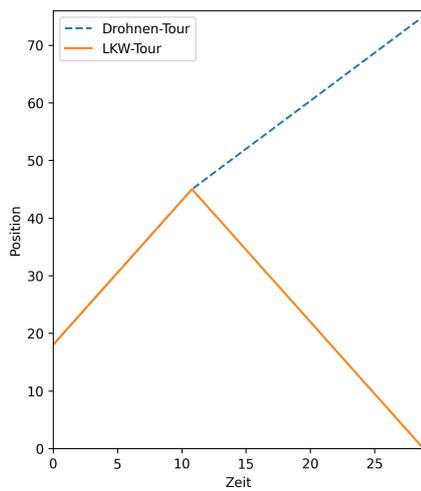
Tabelle 4.1: Parameter und optimale Lösung

Für die angegebene optimale Lösung von Instanz 1 ist der Trennungsort bei $45 \in [p_{n-1}, p_n)$, denn es gilt

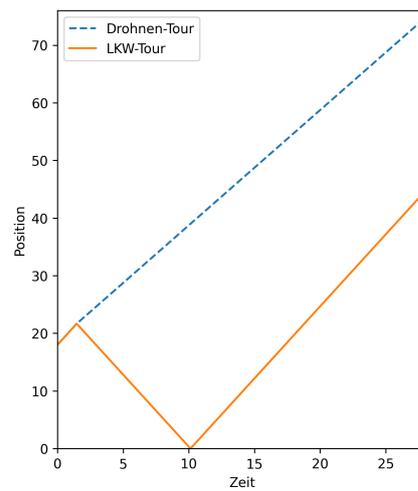
$$\min \max_{i \in \{1, \dots, n\}} \{t_i\} = \min \left\{ \begin{array}{l} 28.8 \quad (x = 45), \\ 32 \quad (x = 29), \\ 34 \quad (x = 4), \\ 51.2 \quad (x = 18), \\ 52.8 \quad , \\ 37.2 \end{array} \right\}.$$

Analog gilt für Instanz 2, dass der Trennungsort bei $x = 65/3 \in [p_{i^*}, p_{n-1}]$ ist, denn es gilt

$$\min \max_{i \in \{1, \dots, n\}} \{t_i\} = \min \left\{ \begin{array}{l} 37.8 \quad (x = 45), \\ 28, 13 \quad (x = 21.67), \\ 34.37 \quad (x = 3.52), \\ 51.2 \quad (x = 18), \\ 52.8 \quad , \\ 37.2 \end{array} \right\}.$$



(a) Optimale Lösung Instanz 1



(b) Optimale Lösung Instanz 2

Abbildung 4.3: Ort-Zeit Diagramm der optimalen Lösungen der beiden Instanzen

Lemma 4.2.7. *Sei ein Line-TSP with Drone-Problem mit $v^D \leq v^T$ und n Kunden mit $1 < i^* < n$ gegeben. Dann ist*

$$\min \max_{i \in \{1, \dots, n\}} \{t_i^m, t_i\} = \min \left\{ \begin{array}{l} \max \left\{ \frac{2 \cdot p_n - 2 \cdot p_1}{v^T + v^D}, \frac{p_{n-1} - 2 \cdot p_1 + p_{i^*}}{v^T} \right\}, \\ \max \left\{ \frac{2 \cdot p_n - 2 \cdot p_1}{v^T + v^D}, \frac{2 \cdot p_n - p_2 - p_{i^*}}{v^T} \right\}, \\ \frac{2p_n - 2p_1}{v^D + v^T} + \frac{p_{i^*} - p_2}{v^T}, \\ \frac{2p_n - 2p_1}{v^D + v^T} + \frac{p_{n-1} - p_{i^*}}{v^T}, \\ \frac{1}{v^T} \cdot (2 \cdot p_n - p_{i^*} - p_1), \\ \frac{1}{v^T} \cdot (-2 \cdot p_1 + p_{i^*} + p_n) \end{array} \right\}$$

Beweis. Nach 4.2.4 existiert eine optimale Lösung, bei der maximal ein Kunde (Kunde 1 oder Kunde n) durch die Drohne bedient wird. Im Gegensatz zu 4.2.4 wird die Drohne entweder sofort gestartet, oder zu Kunde $n - 1$ transportiert, damit diese Kunde n bedienen kann (bzw. zu Kunde 2 transportiert, damit die Drohne Kunde 1 bedienen kann), falls sie einen Kunden bedient. Zunächst wird vorausgesetzt, dass die Drohne Kunde n bedient (für Kunde 1 analog) und der LKW fährt zunächst Richtung Kunde n , startet die Drohne bei $z \in [p_{i^*}, p_{n-1})$ und bedient anschließend die Kunden $i^* - 1, \dots, 2, 1$. Der LKW fährt anschließend sofort Richtung Kunde $n - 1$, um diesen zu bedienen. Es wird vorausgesetzt, dass Drohne und LKW sich erst begegnen, nachdem die Drohne Kunde n bedient hat. Man erhält die folgende LKW-Tour:

$$f_z^T(x) = \begin{cases} v^T x + p_{i^*} & \text{für } x \in \left[0, \frac{z - p_{i^*}}{v^T}\right] \\ -v^T \cdot x + 2 \cdot z - p_{i^*} & \text{für } x \in \left(\frac{z - p_{i^*}}{v^T}, \frac{p_1 - 2 \cdot z + p_{i^*}}{-v^T}\right] \\ v^T \cdot x + 2 \cdot p_1 - 2 \cdot z + p_{i^*} & \text{für } x \in \left(\frac{p_1 - 2 \cdot z + p_{i^*}}{-v^T}, \infty\right) \end{cases}$$

Mit

$$A = \left[\frac{(z - p_{i^*})}{v^T}, \frac{p_n - z \cdot \left(1 - \frac{v^D}{v^T}\right) - p_{i^*} \cdot \frac{v^D}{v^T}}{v^D} \right]$$

$$B = \left(\frac{p_n - z \cdot \left(1 - \frac{v^D}{v^T}\right) - p_{i^*} \cdot \frac{v^D}{v^T}}{v^D}, \infty \right)$$

erhält man die folgende Drohntour:

$$f_z^D(x) = \begin{cases} v^D \cdot x + z \cdot \left(1 - \frac{v^D}{v^T}\right) + p_{i^*} \cdot \frac{v^D}{v^T} & \text{für } x \in A \\ -v^D \cdot x + 2p_n - z \cdot \left(1 - \frac{v^D}{v^T}\right) - p_{i^*} \cdot \frac{v^D}{v^T} & \text{für } x \in B \end{cases}$$

Drohne und LKW begegnen sich demnach zum Zeitpunkt

$(2p_n - 2p_1 + z \cdot (1 + \frac{v^D}{v^T}) - p_{i^*} \cdot (1 + \frac{v^D}{v^T})) \cdot (v^T + v^D)^{-1}$ und der LKW erreicht Kunde $n - 1$ zum Zeitpunkt $(p_{n-1} - 2p_1 + 2z - p_{i^*}) \cdot (v^T)^{-1}$. Man erhält also einen Zielfunktionswert von

$$\max \left\{ \frac{2p_n - 2p_1 + z \cdot (1 + \frac{v^D}{v^T}) - p_{i^*} \cdot (1 + \frac{v^D}{v^T})}{v^T + v^D}, \frac{p_{n-1} - 2p_1 + 2z - p_{i^*}}{v^T} \right\}. \quad (4.1)$$

Für $z \in [p_{i^*}, p_{n-1})$ ist (4.1) für $z = p_{i^*}$ minimal. Analog dazu wird der Fall für den Trennungsort $z \in (p_{n-1}, p_n)$ behandelt. Der Begegnungszeitpunkt von Drohne und LKW ist

$$\frac{2p_n - 2p_1}{v^T + v^D} + \frac{z - p_{i^*}}{v^T}.$$

Wird hingegen die Drohne genau bei Kunde $n - 1$ gestartet, so ist der Begegnungszeitpunkt

$$\frac{2p_n - 2p_1}{v^T + v^D} + \frac{p_{n-1} - p_{i^*}}{v^T}.$$

Soll die Drohne einen Kunden bedienen, so wird sie somit sofort gestartet, oder zum Ort p_2 oder p_{n-1} gebracht, oder die Drohne bedient gar keinen Kunden. \square

Beispiel 4.2.8. Auch hier werden zwei Instanzen präsentiert. In der optimalen Lösung von Instanz 1 wird die Drohne zu Beginn bei p_{i^*} gestartet, um den Kunden n zu bedienen. In der optimalen Lösung von Instanz 2 fährt der LKW zunächst zum Ort p_{n-1} , um die Drohne dort zu starten, damit diese Kunde n bedienen kann. In beiden Instanzen gibt es 5 Kunden. In Instanz 1 ist $v^D = 1$, $v^T = 2$ und $i^* = 3$ und in Instanz 2 ist $i^* = 3$, $v^D = 2$ und $v^T = 2,5$. Die weiteren Parameter und die jeweils optimalen Lösungen sind 4.2 zu entnehmen. In Abbildung 4.4 sind diese optimalen Lösungen dargestellt.

i	1	2	3	4	5	i	1	2	3	4	5
p_{i^*}	0	30	50	55	90	p_i	0	4	30	35	45
t_i	25.0	10.0	0.0	52.5	40.0	t_i	16.0	14.4	0.0	2.0	7.0
r_i^m	-	-	-	-	70	r_i^m	-	-	-	-	15
t_i^m	-	-	-	-	60	t_i^m	-	-	-	-	22
r_i^s	-	-	-	-	50	r_i^s	-	-	-	-	35
t_i^s	-	-	-	-	0	t_i^s	-	-	-	-	2

(a) Instanz 1

(b) Instanz 2

Tabelle 4.2: Parameter und optimale Lösung

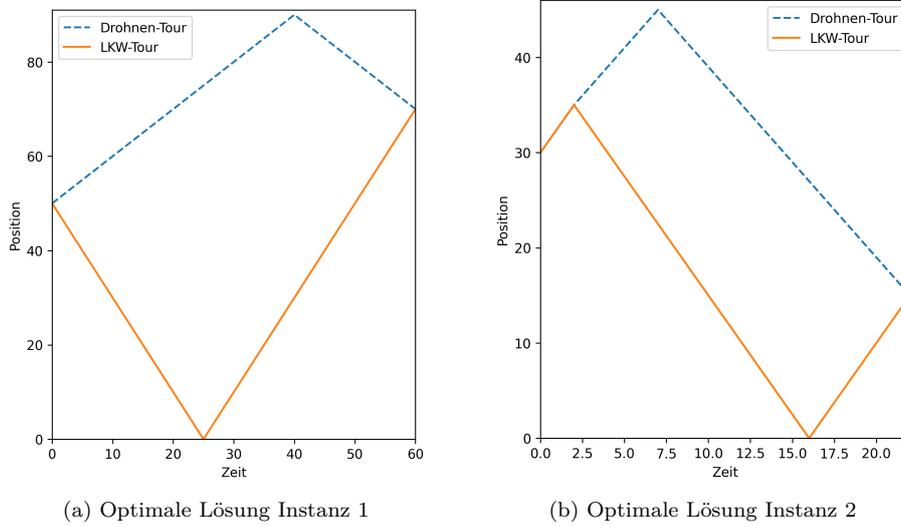


Abbildung 4.4: Ort-Zeit Diagramm der optimalen Lösungen der beiden Instanzen

Für Instanz 1 ergibt sich der optimale Zielfunktionswert gemäß Lemma 4.2.7 wie folgt

$$\min\{60, 60, 75, 62.5, 65, 70\} = 60$$

und für Instanz 2

$$\min\{26, 22.4, 30.4, 22, 24, 30\} = 22.$$

Korollar 4.2.9. *Ist die Drohne langsamer als der LKW und liegen keine Release Dates und Deadlines vor, so erhält man in konstanter Zeit einen optimalen Zielfunktionswert für das Minimieren von $\max_{i \in \{1, \dots, n\}} \{t_i\}$ und das Minimieren von $\max_{i \in \{1, \dots, n\}} \{t_i, t_i^m\}$.*

Beweis. Die Berechnungen in 4.2.4 und 4.2.7 liegen in $\mathcal{O}(1)$. \square

Korollar 4.2.10. *Ist die Drohne langsamer als der LKW und liegen keine Release Dates und Deadlines vor, so lässt sich in linearer Zeit eine optimale Lösung für das Minimieren von $\max_{i \in \{1, \dots, n\}} \{t_i\}$ und das Minimieren von $\max_{i \in \{1, \dots, n\}} \{t_i, t_i^m\}$ ermitteln.*

Beweis. Eine optimale Lösung lässt sich mithilfe der Ausdrücke aus 4.2.4 und 4.2.7 rekonstruieren.

Im Folgenden wird die Lösung für das Minimieren von $\max_{i \in \{1, \dots, n\}} \{t_i\}$ konstruiert, falls für den optimalen Zielfunktionswert z^* gilt

$$\min_{x \in [p_{n-1}, p_n]} \left\{ \max \left\{ \begin{array}{l} \frac{1}{v^T} \cdot (2 \cdot x - p_{i^*} - p_1), \\ \frac{1}{v^D} \cdot (p_n - x) + \frac{1}{v^T} \cdot (x - p_{i^*}) \end{array} \right\} \right\} = z^*.$$

In diesem Fall bringt der LKW die Drohne zunächst zum Ort $x \in [p_{n-1}, p_n]$ und bedient währenddessen die Kunden $i^*, i^* + 1, \dots, n - 1$. Anschließend fährt

der LKW direkt zurück und bedient die Kunden $i^*, i^* - 1, \dots, 1$, während die Drohne Kunde n bedient. Man erhält also den LKW-Operationsvektor

$$a = (i^* + n, i^*, i^* + 1, i^* + 2, \dots, n - 2, n - 1, 2 \cdot n, i^* - 1, i^* - 2, \dots, 2, 1)$$

und die folgenden Fertigstellungszeitpunkte:

$$\begin{aligned} t_{i^*} &= 0 & t_n &= \frac{1}{v^D}(p_n - x) \\ t_{i^*+j} &= \frac{1}{v^T}(p_{i^*+j} - p_{i^*}) & \forall j &\in \{1, \dots, n - 1 - i^*\} \\ t_{i^*-i} &= \frac{1}{v^T}(x - p_i) & \forall i &\in \{1, \dots, i^* + 1\}. \quad \square \end{aligned}$$

An dieser Stelle sei noch angemerkt, dass die Probleme, die mit einer langsameren Drohne und dem Fehlen von Zeitfenstern einhergehen, weiterhin polynomiell lösbar sind, auch wenn der LKW zu seinem Startort zurückkehren muss. Für die zu minimierende Zielfunktion $\max\{t_i, t_i^m\}$ lässt sich weiterhin festhalten, dass die Drohne maximal einen Kunden bedient. Dies kann entweder Kunde 1 oder Kunde n sein. Die Orte der Kunden $1, 2, i^*, n - 1$ und n sind dabei relevant. Unter den nachfolgend dargestellten partiell zulässigen Vektoren findet sich einer, der zu einer optimalen Lösung führt:

$$\begin{aligned} (i^*, i^* + 2n, n - 1, 2n, 1, 3n, Depot) & & (i^*, i^* + 2n, 2n, 1, 3n, n - 1, Depot) \\ (i^*, i^* + 2n, 2n, 1, n - 1, 3n, Depot) & & (i^*, i^* + 2n, 1, n, Depot) \\ (i^*, i^* + 2n, 2, n + 1, n, 2n + 1, Depot) & & (i^*, i^* + 2n, n + 1, n, 2, 2n + 1, Depot) \\ (i^*, i^* + 2n, n + 1, n, 2n + 1, 2, Depot) & & (i^*, i^* + 2n, n, 1, Depot). \end{aligned}$$

Im Falle einer Minimierung von $\max\{t_i\}$ existiert eine optimale Lösung, bei der die Drohne maximal zwei Kunden bedient. Dabei kann es sich um die Kombinationen 1 und n , 1 und 2 oder $n - 1$ und n handeln. In analoger Weise lassen sich die partiell zulässigen Vektoren angeben, die eine optimale Lösung enthalten. Anschließend kann das Lineare Programm aus Abschnitt 3.2.3 genutzt werden, um eine optimale Lösung zu finden.

4.3 Zeitfenster und $v^D = 0$ oder $v^T = 0$

In diesem Abschnitt werden die Fälle $v^D = 0$ und $v^T = 0$ betrachtet. Es wird sich zeigen, dass ein 1-Maschinen-Scheduling-Problem vorliegt, falls $v^T = 0$ gilt, da lediglich die Drohne für die Kunden verfügbar ist. Ist hingegen $v^D = 0$ und sind lediglich Release Dates oder Deadlines gegeben, so handelt es sich um das Line-TSP, wie in (Tsitsiklis, 1992) beschrieben. Das Line-TSPwD kann daher als eine Verallgemeinerung dieser Probleme angesehen werden.

Lemma 4.3.1. *Sei eine Instanz mit $v^T = 0$ und n Kunden gegeben, wobei jedem Kunden i ein Release Date r_i zugewiesen ist. Dann gibt es einen polynomiellen Algorithmus, der das Line-TSPwD mit dem Ziel der Minimierung von $\max\{t_i^m, t_i\}$ löst.*

Beweis. Es wird vorausgesetzt, dass $T = \emptyset$ ist, da sonst keine zulässige Lösung existieren würde. Bei einer Geschwindigkeit des LKWs von 0 muss die Drohne

alle Kunden bedienen, während der LKW als Depot dient. Für einen gegebenen Schedule σ lässt sich der Begegnungszeitpunkt von Drohne und LKW, nachdem der letzte Kunde bedient wurde, wie folgt berechnen:

$$t_{\sigma(n)}^m = \max \left\{ r_{\sigma(n)} + \frac{|p_{\sigma(n)} - p_{i^*}|}{v^D}, t_{\sigma(n-1)}^m + 2 \frac{|p_{\sigma(n)} - p_{i^*}|}{v^D} \right\},$$

wobei $t_{\sigma(0)}^m$ gleich 0 definiert wird. Ein optimaler Schedule lässt sich konstruieren, wenn unter allen noch nicht bedienten Kunden aus $S \subseteq \{1, \dots, n\}$ zunächst Kunde j bedient wird mit

$$r_j - \frac{|p_j - p_{i^*}|}{v^D} = \min_{i \in S} \left\{ r_i - \frac{|p_i - p_{i^*}|}{v^D} \right\}.$$

Sei σ ein optimaler Schedule, der diese Vorschrift nicht erfüllt. Dann gibt es ein $k \in \{1, \dots, n\}$ mit

$$r_{\sigma(k)} - \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} > r_{\sigma(k+1)} - \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D}. \quad (4.2)$$

1. Fall: Es gelte

$$r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} > t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D}. \quad (4.3)$$

Dies impliziert, dass die Drohne bei Kunde $\sigma(k)$ warten muss und somit der Begegnungszeitpunkt $t_{\sigma(k)}^m = r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D}$ gilt. Hieraus resultiert die folgende Abschätzung:

$$\begin{aligned} t_{\sigma(k)}^m + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} &> r_{\sigma(k)} - \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &\stackrel{(4.2)}{>} r_{\sigma(k+1)} - \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &= r_{\sigma(k+1)} + \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D}. \end{aligned}$$

Also ist $t_{\sigma(k+1)}^m = t_{\sigma(k)}^m + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D}$. Sei σ' der Schedule, der entsteht, wenn $\sigma(k)$ und $\sigma(k+1)$ in σ getauscht werden, also

$$\sigma' = (\sigma(1), \dots, \sigma(k-1), \sigma(k+1), \sigma(k), \sigma(k+2), \dots, \sigma(n)).$$

Dann ist

$$t_{\sigma'(k+1)}^m = \max\{a, \max\{b, c\} + d\},$$

mit

$$\begin{aligned} a &= r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} & b &= r_{\sigma(k+1)} + \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ c &= t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} & d &= 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D}. \end{aligned}$$

Es gelten die folgenden Abschätzungen:

$$t_{\sigma(k+1)}^m = r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} > r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} = a$$

$$\begin{aligned} t_{\sigma(k+1)}^m &= r_{\sigma(k)} - \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} \\ &\stackrel{(4.2)}{>} r_{\sigma(k+1)} - \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} \\ &= r_{\sigma(k+1)} + \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} = b + d \end{aligned}$$

$$\begin{aligned} t_{\sigma(k+1)}^m &= r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &= r_{\sigma(k)} - \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &\stackrel{(4.3)}{>} t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} = c + d. \end{aligned}$$

Also ist $t_{\sigma(k+1)}^m > t_{\sigma'(k+1)}^m$ und der Begegnungszeitpunkt des letzten Kunden nach der Vertauschung der beiden Kunden ist in σ' nicht später als in σ .

2. Fall: Nun gelte

$$r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} \leq t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D}. \quad (4.4)$$

Das bedeutet, dass die Drohne nicht beim Kunden $\sigma(k)$ warten muss und folglich der Begegnungszeitpunkt $t_{\sigma(k)}^m = t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D}$ resultiert. Dies führt zur folgenden Abschätzung:

$$\begin{aligned} t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} &> t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &\stackrel{(4.4)}{\geq} r_{\sigma(k)} - \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &\stackrel{(4.2)}{>} r_{\sigma(k+1)} - \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &= r_{\sigma(k+1)} + \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D}. \end{aligned}$$

Somit ist $t_{\sigma(k+1)}^m = t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D}$.

Nun wird wieder der Schedule σ' analysiert, der entsteht, wenn in σ die Kunden $\sigma(k)$ und $\sigma(k+1)$ getauscht werden. Es gilt

$$t_{\sigma'(k+1)}^m = \max \{a, \max\{b, c\} + d\}$$

mit

$$a = r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} \qquad b = r_{\sigma(k+1)} + \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D}$$

$$c = t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \quad d = 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D}.$$

Der Begegnungszeitpunkt $t_{\sigma(k+1)}^m$ lässt sich wie folgt nach unten abschätzen:

$$\begin{aligned} t_{\sigma(k+1)}^m &= t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &\stackrel{(4.4)}{\geq} r_{\sigma(k)} - \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &= r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} > a \end{aligned} \quad (4.5)$$

$$\begin{aligned} t_{\sigma(k+1)}^m &\stackrel{(4.5)}{\geq} r_{\sigma(k)} + \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &= r_{\sigma(k)} - \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &\stackrel{(4.2)}{>} r_{\sigma(k+1)} - \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} \\ &= r_{\sigma(k+1)} + \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} = b + d \end{aligned}$$

$$t_{\sigma(k+1)}^m = t_{\sigma(k-1)}^m + 2 \frac{|p_{\sigma(k)} - p_{i^*}|}{v^D} + 2 \frac{|p_{\sigma(k+1)} - p_{i^*}|}{v^D} = c + d.$$

Insgesamt ist also $t_{\sigma(k+1)}^m \geq t_{\sigma'(k+1)}^m$. Jeder Schedule kann durch eine endliche Hintereinanderausführung solcher Auftragsvertauschungen in einen Schedule überführt werden, der die obige Vorschrift erfüllt, ohne dass sich der Zielfunktionswert verschlechtert. Dies gilt auch für optimale Schedules. Eine optimale Lösung kann in $\mathcal{O}(n + n \log(n))$ vielen Rechenschritten gefunden werden, da für alle $i \in \{1, \dots, n\}$ der Ausdruck $r_i - \frac{|p_i - p_{i^*}|}{v^D}$ berechnet und einsortiert werden muss. \square

Lemma 4.3.2. *Sei eine Instanz mit $v^T = 0$ und n Kunden gegeben, wobei jedem Kunden i ein Release Date r_i zugewiesen ist. Dann existiert ein polynomieller Algorithmus, der das Line-TSPwD mit dem Ziel der Minimierung von $\max\{t_i\}$ löst.*

Beweis. In diesem Fall fliegt die Drohne nicht zurück zum LKW, nachdem diese den letzten Kunden bedient hat. Hier liegen also n Möglichkeiten vor, welcher Kunde zuletzt bedient wird. Die übrigen Kunden werden gemäß 4.3.1 bedient, man ermittelt also

$$\min_{j \in \{1, \dots, n\}} \max \left\{ r_j, \left(\min_{i \in \{1, \dots, n\} \setminus \{j\}} \max\{t_i^m, t_i\} \right) + \frac{|p_j - p_{i^*}|}{v^D} \right\}. \quad \square$$

Lemma 4.3.3. *Sei eine Instanz mit $v^T = 0$ und n Kunden gegeben, wobei jedem Kunden i eine Deadline d_i zugewiesen ist. Dann gibt es einen polynomiellen Algorithmus, der das Line-TSPwD mit dem Ziel der Minimierung von $\max\{t_i^m, t_i\}$ Problem löst.*

Beweis. Da auch hier der LKW nur als unbewegliches Depot dient, muss die Drohne alle Kunden bedienen. Dieses Problem lässt sich als ein 1-Maschinen-Scheduling Problem mit dem Ziel der Minimierung der maximalen Verspätung auffassen (abgekürzt $1 || L_{\max}$ gemäß 3-Feld-Notation in (Graham, Lawler, Lenstra & Kan, 1979)), wobei aber $L_{\max} \leq 0$ sein muss. Andernfalls würde das Problem keine zulässige Lösung besitzen. Die Deadlines lauten

$$d_i^{neu} = d_i + \frac{|p_i - p_{i^*}|}{v^D}$$

und die Prozesszeiten sind

$$\rho_i = 2 \frac{|p_i - p_{i^*}|}{v^D}$$

Auch dieses Problem kann mittels einer einfachen Prioritätenregel gelöst werden, der sogenannten *Earliest Due Date-Regel* (EDD). Diese Regel ist in der Literatur unter anderem in (Pinedo, 2008)(S.44) beschrieben. \square

Lemma 4.3.4. *Sei eine Instanz mit $v^T = 0$ und n Kunden gegeben, wobei jedem Kunden i eine Deadline d_i zugewiesen ist. Dann gibt es einen polynomiellen Algorithmus, der das Line-TSPwD mit dem Ziel der Minimierung von $\max\{t_i\}$ Problem löst.*

Beweis. Die ersten $n - 1$ Kunden werden gemäß 4.3.3 bedient und der letzte Kunde wird bedient, ohne dass anschließend die Drohne zurückfliegt. Es wird also eine zulässige Lösung gesucht mit

$$\min_{j \in \{1, \dots, n\}} \left(\sum_{i \in \{1, \dots, n\} \setminus \{j\}} 2 \frac{|p_i - p_{i^*}|}{v^D} \right) + \frac{|p_j - p_{i^*}|}{v^D}.$$

Die Kunden in $\{1, \dots, n\} \setminus \{j\}$ werden gemäß 4.3.3 bedient und anschließend Kunde j .

Eine mögliche Vorgehensweise, um eine optimale Lösung zu ermitteln, wäre zunächst EDD gemäß Lemma 4.3.3 auf die Kundenmenge $\{1, \dots, n\}$ anzuwenden. Ist diese Lösung zulässig, so wird eine optimale Lösung des Problems

$$\begin{aligned} & \max_{j \in \{1, \dots, n\}} y_j \cdot |p_j - p_{i^*}| \\ \text{sodass } & y_j \left(d_j + \left(\frac{|p_j - p_{i^*}|}{v^D} \right) \right) \geq y_j \sum_{i=1}^n \frac{|p_i - p_{i^*}|}{v^D} \quad \forall j \in \{1, \dots, n\} \quad (4.6) \\ & \sum_{i=1}^n y_i = 1 \quad y_i \in \{0, 1\} \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

gesucht. Um eine optimale Lösung hierfür zu finden, werden $\mathcal{O}(n)$ viele Rechenschritte benötigt. Für alle $j \in \{1, \dots, n\}$ wird getestet, ob die Restriktion (4.6) erfüllt ist. Anschließend wird unter diesen der Kunde mit maximalem $|p_j - p_{i^*}|$ ausgewählt. \square

Lemma 4.3.5. *Sei eine Instanz mit $v^D = 0$ und n Kunden gegeben, wobei jedem Kunden i eine Deadline d_i zugewiesen ist. Dann gibt es einen polynomiellen Algorithmus, der das Line-TSPwD löst.*

Beweis. Hat die Drohne eine Geschwindigkeit von 0 und liegen ausschließlich Deadlines vor, so gibt es eine optimale Lösung, in der die Drohne keinen Kunden bedient, sowohl für die Minimierung von $\max\{t_i^m, t_i\}$, als auch für die Minimierung von $\max\{t_i\}$. Angenommen die Drohne bedient einen Kunden, so muss der LKW die Drohne bis zu diesem Kunden fahren. Da nun der LKW aber zum Kunden fährt und keine Release Dates vorliegen, kann er auch den Kunden selbst bedienen.

Eine Instanz, in der ausschließlich der LKW die Kunden bedient und Deadlines existieren, kann als eine Line-TSPTW-Instanz mit Deadlines aufgefasst werden. Diese Instanzen können mithilfe der Dynamischen Programmierung gelöst werden. In (Tsitsiklis, 1992) (Theorem 1) ist ein polynomieller Ansatz zu finden. \square

Lemma 4.3.6. *Sei eine Instanz mit $v^D = 0$ und n Kunden gegeben, wobei jedem Kunden i ein Release Date r_i zugewiesen ist. Dann gibt es einen polynomiellen Algorithmus, der das Line-TSP-with-Drone Problem mit dem Ziel der Minimierung von $\max\{t_i^m, t_i\}$ löst.*

Beweis. Auch in diesem Fall gibt es eine optimale Lösung, in der die Drohne keinen Kunden bedient. Würde die Drohne einen Kunden bedienen, so muss der LKW die Drohne wieder auflesen. Da dies nicht vor dem Release Date geschieht, kann der LKW den Kunden bedienen. Somit liegt ein Line-TSP mit Release Dates vor. Auch dieses Problem kann polynomiell mithilfe der Dynamischen Programmierung gelöst werden (Psaraftis et al., 1990) (Theorem 2.2). \square

Lemma 4.3.7. *Sei eine Instanz mit $v^D = 0$ und n Kunden gegeben, wobei jedem Kunden i ein Release Date r_i zugewiesen ist. Dann gibt es einen polynomiellen Algorithmus, der das Line-TSPwD mit dem Ziel der Minimierung von $\max\{t_i\}$ löst.*

Beweis. Es gibt eine optimale Lösung, in der die Drohne maximal einen Kunden bedient. Würde die Drohne mehr als einen Kunden bedienen, so muss der LKW sie wieder auflesen und kann diesen Kunden selbst bedienen. Nur nachdem der letzte Kunde durch die Drohne bedient wurde, muss der LKW diese nicht wieder auflesen. Sei k der Kunde, der durch die Drohne bedient wird. Das Problem wird nun als ein Line-TSP ohne Drohne aufgefasst. Das Release Date des Kunden k wird auf 0 gesetzt, da die Drohne zu jedem beliebigen Zeitpunkt dort abgesetzt werden kann. Eine optimale Lösung mit der Voraussetzung, dass die Drohne Kunde k bedient, lässt sich mithilfe des Ansatzes in (Psaraftis et al., 1990) lösen. Sei $z(k)$ der Zielfunktionswert einer optimalen Lösung des Line-TSP, wobei Kunde k das neue Release Date $r_k^{neu} = 0$ hat. Dann ist $\max\{z(k), r_k\}$ der Zielfunktionswert einer optimalen Lösung, bei der die Drohne Kunde k bedient. Insgesamt erhält man also den optimalen Zielfunktionswert $\min_{k \in \{1, \dots, n\}} \max\{r_k, z(k)\}$. Es wird also der Ansatz von (Psaraftis et al., 1990) n Mal für unterschiedliche Kunden k angewendet. \square

Sofern lediglich Release Dates oder Deadlines vorliegen und die Drohne langsamer als der LKW ist, kann das Problem im Fall von $v^T = 0$ mit einfachen Prioritätsregeln gelöst werden. Im Fall $v^D = 0$ lässt es sich polynomiell mithilfe der dynamischen Programmierung lösen. Liegen jedoch Release Dates und Deadlines vor, so sind auch in diesen Fällen die Probleme \mathcal{NP} -schwer. Ist $v^T = 0$, so liegt ein 1-Maschinen Scheduling Problem vor, bei dem jeder Auftrag eine individuelle Prozesszeit hat und diese Prozesszeit in einem vorgegebenen Zeitfenster

stattfinden muss. In (Garey & Johnson, 1979) (Theorem 4.5) ist eine pseudo-polynomielle Transformation von 3-Partition auf dieses Problem zu finden. Ist $v^T = 0$ und $T = \{1, 2, \dots, n\}$, so liegt das Line-TSP mit Zeitfenstern vor, welches ebenfalls \mathcal{NP} -schwer ist. Dies kann (Tsitsiklis, 1992) (Theorem 2) entnommen werden. Dort wird eine Reduktion einer Instanz von MSAT auf das Line-TSP mit Zeitfenstern konstruiert.

4.4 Keine Zeitfenster und $v^D > v^T$

In diesem Abschnitt werden zunächst Fälle behandelt, in denen der LKW aufgrund der Struktur der Instanz konstant in eine Richtung fährt, oder eine bestimmte LKW-Tour vorausgesetzt wird. Eine solche vorgegebene LKW-Tour entspricht zwar nicht der Definition des Line-TSPwD, jedoch werden diese Probleme zu einem späteren Zeitpunkt für die Bestimmung der Komplexität gebraucht.

Sei nun die Drohne schneller, als der LKW und $i^* \in \{1, n\}$ der Startkunde (o.B.d.A. $i^* = 1$). Die Drohne ist ununterbrochen aktiv. Sie bedient einen Kunden und fliegt wieder zurück zum LKW, bis alle Kunden bedient wurden, während der LKW in Richtung Kunde n fährt. Zunächst wird der Fall bezüglich der Zielfunktion $\min \max\{t_i^m, t_i\}$ und anschließend bezüglich der Zielfunktion $\min \max\{t_i\}$ behandelt. Dazu ist im Wesentlichen nur noch die Reihenfolge der Kunden zu bestimmen, die durch die Drohne bedient werden. Zunächst wird ein Ausdruck in geschlossener Form des letzten Begegnungszeitpunkts für eine gegebene Kundenreihenfolge bestimmt, die durch die Drohne bedient werden.

Definition 4.4.1. Seien n Kunden auf einer Linie gegeben, die durch eine Drohne bedient werden sollen, sowie eine feste LKW-Tour. Ferner sei $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ eine Reihenfolgefestlegung der Kunden, die durch die Drohne bedient werden. Der Schedule $(\sigma(1), \dots, \sigma(n))$ wird als *Drohnen-Schedule* bezeichnet.

Anmerkung 4.4.2. Um einen Ausdruck für den Begegnungszeitpunkt von Drohne und LKW zu ermitteln, werden zunächst die Begegnungszeitpunkte für den Drohnen-Schedule $(\sigma(1), \sigma(2), \dots, \sigma(n))$ ermittelt. LKW und Drohne starten zum Zeitpunkt 0 am Ort x_0 . Es gilt $x_0 < p_j$ für alle $j \in \{1, \dots, n\}$ und es wird vorausgesetzt, dass der LKW keinen der Kunden $1, \dots, n$ in dieser Zeit erreicht. Außerdem gilt $t_1^s = 0$. Abbildung 4.5 veranschaulicht die nachfolgenden Berechnungen. Es wird gezeigt, dass

$$\begin{aligned}
 t_{\sigma(1)} &= \frac{p_{\sigma(1)} - x_0}{v^D} & t_{\sigma(1)}^m &= \frac{2 \cdot p_{\sigma(1)} - 2 \cdot x_0}{v^T + v^D} \\
 r_{\sigma(1)}^m &= v^T \cdot \frac{2 \cdot p_{\sigma(1)} - 2 \cdot x_0}{v^T + v^D} + x_0 & t_{\sigma(i)} &= t_{\sigma(i-1)}^m + \frac{p_{\sigma(i)} - r_{\sigma(i-1)}^m}{v^D} \\
 t_{\sigma(i)}^m &= \frac{p_{\sigma(i)} + v^D \cdot t_{\sigma(i)} - x_0}{v^D + v^T} & r_{\sigma(i)}^m &= v^T \cdot t_{\sigma(i)}^m + x_0
 \end{aligned} \tag{4.7}$$

gelten. Die Begegnungs- und Fertigstellungszeiten sind Schnittpunkte von Geraden. Sei f^T die Tour des LKWs, $f_{(1, \sigma(1))}^D$ die Teiltour der Drohne, bei der sie zum Zeitpunkt $t_{\sigma(1)}^s$ den LKW verlässt, um Kunde $\sigma(1)$ zu bedienen, und

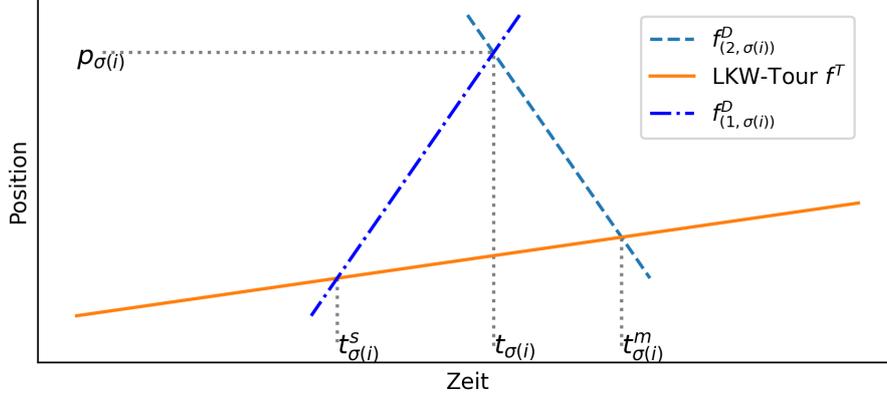


Abbildung 4.5: Begegnungs- und Trennungszeitpunkte

$f_{(2, \sigma(1))}^D$ die Teiltour der Drohne, bei welcher sie Kunde $\sigma(1)$ zum Zeitpunkt $t_{\sigma(1)}$ verlässt und zum LKW zurückkehrt. Diese Teiltour der Drohne ist

$$f_{(2, \sigma(1))}^D(t) = -v^D t + 2p_{\sigma(1)} - x_0,$$

da sich die Drohne zum Zeitpunkt $t_{\sigma(1)}$ am Ort $p_{\sigma(1)}$ befindet:

$$f_{(2, \sigma(1))}^D(t_{\sigma(1)}) = -v^D \left(\frac{p_{\sigma(1)} - x_0}{v^D} \right) + (2p_{\sigma(1)} - x_0) = p_{\sigma(1)}.$$

Die LKW-Tour ist definiert durch $f^T(t) = v^T t + x_0$. Also ist

$$-v^D t_{\sigma(1)}^m + 2p_{\sigma(1)} - x_0 = v^T t_{\sigma(1)}^m + x_0 \iff t_{\sigma(1)}^m = \frac{2p_{\sigma(1)} - 2x_0}{v^T + v^D}.$$

Ebenso wird $t_{\sigma(i)}^m$ ermittelt. Hier ist $f_{(2, \sigma(i))}^D(t) = -v^D t + p_{\sigma(i)} + v^D t_{\sigma(i)}$ und somit

$$-v^D t_{\sigma(i)}^m + p_{\sigma(i)} + v^D t_{\sigma(i)} = v^T t_{\sigma(i)}^m + x_0 \iff t_{\sigma(i)}^m = \frac{p_{\sigma(i)} + v^D t_{\sigma(i)} - x_0}{v^D + v^T}.$$

$f_{(1, \sigma(i))}^D$ in Abbildung 4.5 entspricht der Drohnen-Tour, um vom LKW zum Kunden $\sigma(i)$ zu fliegen.

Für eine Instanz, bei welcher der LKW eine Tour der Form $f^T(t) = v^T \cdot t + x_0$ oder $f^T(t) = -v^T \cdot t + x_0$ fährt und das Ziel die Minimierung von $\max\{t_i^m, t_i\}$ ist, lässt sich als 1-Maschinen Scheduling Problem mit startzeitabhängigen Prozesszeiten auffassen, wobei die Prozesszeiten linear mit dem Startzeitpunkt wachsen oder fallen. Theorem 2 aus (Gupta & Gupta, 1988) besagt, dass für ein 1-Maschinen Scheduling Problem mit den Aufträgen $1, \dots, n$ mit Prozesszeiten $p_i(t) = a_i + b_i \cdot t$ und $a_i/b_i \leq a_{i+1}/b_{i+1}$ für alle $i \in \{1, \dots, n-1\}$ der Schedule $(n, n-1, \dots, 2, 1)$ eine minimale Bearbeitungsdauer hat. Liegt die LKW-Tour f^T vor, so ist der Begegnungszeitpunkt des k -ten Jobs, der durch die Drohne

bedient wird in Abhängigkeit des Trennungzeitpunkts t gegeben durch

$$\frac{v^D - v^T}{v^D + v^T} \cdot t + \frac{2p_k - 2x_0}{v^D + v^T}.$$

Da nun die Drohne schneller als der LKW ist, gilt für $p_k > p_{k-1}$ die Gleichung

$$\frac{2p_k - 2x_0}{v^D + v^T} \cdot \left(\frac{v^D - v^T}{v^D + v^T} \right)^{-1} = \frac{2p_k - 2x_0}{v^D - v^T} \geq \frac{2p_{k-1} - 2x_0}{v^D - v^T}.$$

Somit lassen sich die noch folgenden Aussagen 4.4.5 und 4.4.16 mittels Theorem 2 aus (Gupta & Gupta, 1988) ableiten.

Lemma 4.4.3. *Sei $v^D > v^T$. Darüber hinaus seien Kunden $1, \dots, n$ gegeben sowie ein Drohnen-Schedule $(\sigma(1), \dots, \sigma(n))$, wobei die Drohne bis zum Zeitpunkt $t_{\sigma(1)}^s$ den LKW nicht verlässt und anschließend ohne Pausen ihre Kunden bedient. Außerdem fährt der LKW strikt in eine Richtung, d.h. die LKW-Tour ist gegeben durch $f^T(t) = v^T \cdot t + x_0$, wobei $x_0 < \min_{i \in \{1, \dots, n\}} p_i$ gilt. Sei zusätzlich $f^T(t_{\sigma(n)}^m) < \min_{i \in \{1, \dots, n\}} p_i$. Dann ist*

$$t_{\sigma(n)} = t_{\sigma(1)}^s \left(1 - \frac{v^T}{v^D} \right) \left(\frac{v^D - v^T}{v^D + v^T} \right)^{n-1} + 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \left(\frac{p_{\sigma(n-k)} - x_0}{v^D} \right) \right) - \frac{p_{\sigma(n)}}{v^D} + \frac{x_0}{v^D}$$

Beweis. Der Beweis verläuft induktiv über die Anzahl der Kunden. Sei $n = 1$. Dann ist

$$\begin{aligned} t_{\sigma(1)} &= \frac{p_{\sigma(1)} - f^T(t_{\sigma(1)}^s)}{v^D} + t_{\sigma(1)}^s = \frac{p_{\sigma(1)} - v^T \cdot t_{\sigma(1)}^s - x_0}{v^D} + t_{\sigma(1)}^s \\ &= t_{\sigma(1)}^s \left(1 - \frac{v^T}{v^D} \right) \left(\frac{v^D - v^T}{v^D + v^T} \right)^{1-1} \\ &\quad + 2 \left(\sum_{k=0}^{1-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \left(\frac{p_{\sigma(n-k)} - x_0}{v^D} \right) \right) - \frac{p_{\sigma(1)} - x_0}{v^D} \end{aligned}$$

Nach 4.4.2 und der Induktionsvoraussetzung gilt somit

$$\begin{aligned} t_{\sigma(n)} &= t_{\sigma(n-1)}^m + \frac{p_{\sigma(n)} - r_{\sigma(n-1)}^m}{v^D} = \frac{p_{\sigma(n-1)} + v^D \cdot t_{\sigma(n-1)} - x_0}{v^D + v^T} \\ &\quad + \frac{p_{\sigma(n)} - v^T \cdot t_{\sigma(n-1)}^m - x_0}{v^D} \\ &= t_{\sigma(n-1)} \left(\frac{v^D - v^T}{v^D + v^T} \right) + \frac{p_{\sigma(n-1)}}{v^D} \left(\frac{v^D - v^T}{v^D + v^T} \right) + \frac{p_{\sigma(n)}}{v^D} - \frac{x_0}{v^D} \left(\frac{v^D - v^T}{v^D + v^T} \right) \\ &\quad - \frac{x_0}{v^D} \\ &\stackrel{\text{I.V.}}{=} \left(\frac{v^D - v^T}{v^D + v^T} \right) \left(t_{\sigma(1)}^s \cdot \left(1 - \frac{v^T}{v^D} \right) \left(\frac{v^D - v^T}{v^D + v^T} \right)^{n-2} \right) \end{aligned}$$

$$\begin{aligned}
& + 2 \left(\sum_{k=0}^{n-2} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \left(\frac{p_{\sigma(n-1-k)} - x_0}{v^D} - \frac{x_0}{v^D} \right) \right) - \frac{p_{\sigma(n-1)}}{v^D} + \frac{x_0}{v^D} \\
& + \frac{p_{\sigma(n-1)}}{v^D} \cdot \left(\frac{v^D - v^T}{v^D + v^T} \right) + \frac{p_{\sigma(n)}}{v^D} - \frac{x_0}{v^D} \cdot \left(\frac{v^D - v^T}{v^D + v^T} \right) - \frac{x_0}{v^D} \\
& = t_{\sigma(1)}^s \left(1 - \frac{v^T}{v^D} \right) \cdot \left(\frac{v^D - v^T}{v^D + v^T} \right)^{n-1} \\
& + 2 \left(\sum_{k=0}^{n-2} \left(\frac{v^D - v^T}{v^D + v^T} \right)^{k+1} \frac{p_{\sigma(n-1-k)} - x_0}{v^D} \right) + \frac{p_{\sigma(n)}}{v^D} + \frac{p_{\sigma(n)}}{v^D} - \frac{p_{\sigma(n)}}{v^D} \\
& - \frac{x_0}{v^D} - \frac{x_0}{v^D} + \frac{x_0}{v^D} \\
& = t_{\sigma(1)}^s \left(1 - \frac{v^T}{v^D} \right) \left(\frac{v^D - v^T}{v^D + v^T} \right)^{n-1} \\
& + 2 \left(\sum_{k=-1}^{n-2} \left(\frac{v^D - v^T}{v^D + v^T} \right)^{k+1} \frac{p_{\sigma(n-1-k)} - x_0}{v^D} \right) - \frac{p_{\sigma(n)}}{v^D} + \frac{x_0}{v^D} \\
& = t_{\sigma(1)}^s \left(1 - \frac{v^T}{v^D} \right) \left(\frac{v^D - v^T}{v^D + v^T} \right)^{n-1} \\
& + 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \frac{p_{\sigma(n-k)} - x_0}{v^D} \right) - \frac{p_{\sigma(n)}}{v^D} + \frac{x_0}{v^D} \quad \square
\end{aligned}$$

Das folgende Lemma ergibt sich analog für die Ermittlung des Begegnungszeitpunkts.

Lemma 4.4.4. *Es liegt die gleiche Situation wie in Lemma 4.4.3 vor. Dann ist*

$$t_{\sigma(n)}^m = t_{\sigma(1)}^s \cdot \left(\frac{v^D - v^T}{v^D + v^T} \right)^n + 2 \cdot \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \cdot \left(\frac{p_{\sigma(n-k)}}{v^D + v^T} - \frac{x_0}{v^D + v^T} \right) \right)$$

Lemma 4.4.5. *Seien n Kunden gegeben mit $p_i > p_{i-1}$. Gilt für die LKW-Tour $f^T(t) = v^T \cdot t + x_0$ die Eigenschaft $f^T(t_1^m) < \min_{i \in \{1, \dots, n\}} p_i$ für Drohnen-Schedule $(n, n-1, \dots, 1)$, so ist die gegebene LKW-Tour und der gegebene Drohnen-Schedule optimal bezüglich $\min \max\{t_i^m, t_i\}$.*

Beweis. Die Drohne bedient die Kunden ab dem Zeitpunkt 0 in der Reihenfolge $\sigma(1), \dots, \sigma(n)$. Der letzte Kunde wird zum Zeitpunkt

$$\begin{aligned}
t_{\sigma(n)}^m & = 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \cdot \left(\frac{p_{\sigma(n-k)}}{v^D + v^T} - \frac{x_0}{v^D + v^T} \right) \right) \\
& = 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \frac{-x_0}{v^D + v^T} \right) + \frac{2}{v^D + v^T} \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k p_{\sigma(n-k)} \right)
\end{aligned}$$

bedient. Wegen Lemma 4.4.4 liegt der Begegnungszeitpunkt danach. Da $\left(\left(\frac{v^D - v^T}{v^D + v^T} \right)^k \right)_{k \in \mathbb{N}}$ eine streng monoton fallende Folge ist und obiger Ausdruck minimiert werden soll, ist die Zuordnung:

$$\sigma(k) = n - k$$

offenbar optimal. \square

Sind also nur Kunden auf der einen Seite zu bedienen, d.h. $i^* \in \{1, n\}$, so fährt der LKW strikt in Richtung x mit $x \in \{1, n\} \setminus \{i^*\}$ und die Drohne bedient den von ihr aus entferntesten Kunden und fliegt anschließend direkt zum LKW, um von dort aus erneut zu starten und nun wieder den entferntesten, bisher noch nicht bedienten Kunden zu bedienen, bis alle Kunden bedient wurden. Dann kehrt sie zurück zum LKW. Dies führt zu einer optimalen Lösung bezüglich $\min \max\{t_i^m, t_i\}$. Dies beschreibt Algorithmus 4.4.6. Hier wird davon ausgegangen, dass bei $i^* = 1$ gestartet wird. Der Parameter l ist der Kunde, der noch nicht vom LKW bedient wurde, wobei aber Kunde $l - 1$ durch den LKW bedient wurde. Zu Beginn ist dies Kunde 2. In der äußeren For-Schleife werden nach und nach die von p_1 entferntesten, noch nicht durch die Drohne bedienten Kunden i gewählt, und es wird überprüft, ob dieser vom LKW schon bedient wurde oder nicht. Sofern der LKW diesen noch nicht bedient hat, d.h. $l \leq i$, bedient die Drohne Kunden i und es wird geprüft, welche Kunden der LKW bis zum Fertigstellungszeitpunkt von Kunde i bedienen kann. Zusätzlich wird überprüft, welche Kunden der LKW zwischen dem Fertigstellungszeitpunkt von Kunde i und dem Begegnungszeitpunkt von Drohne und LKW, nach dem die Drohne Kunde i bedient hat, bedienen kann.

Eine Lösung lässt sich als Vektor darstellen, der aus den folgenden Einträgen besteht:

Eintrag	Bedeutung
(i, t_i)	Der LKW bedient Kunde i zum Zeitpunkt t_i
(s, i, t_i^s, r_i^s)	Die Drohne wird zum Zeitpunkt t_i^s am Ort r_i^s gestartet, um Kunde i zu bedienen
(v, i, t_i)	Die Drohne bedient zum Zeitpunkt t_i Kunde i
(m, i, t_i^m, r_i^m)	Drohne und LKW begegnen sich zum Zeitpunkt t_i^m bei r_i^m , nachdem die Drohne Kunde i bedient hat

Algorithmus 4.4.6

Data: Kunden $i \in \{1 = i^*, \dots, n\}$ mit $p_i < p_{i+1}$ und Geschwindigkeiten v^D und v^T

Result: Optimale Lösung für die Minimierung von $\max\{t_i^m, t_i\}$.

Initialize: $t_1 = 0, t_i = \infty \forall i \in \{2, \dots, n\}, l = 2, k = 2, List = [(1, 0)],$
 $t_{n+1}^m = 0, t_i^m = \infty \forall i \in \{1, \dots, n\}, t_i^s = \infty \forall i \in \{1, \dots, n\}$
 $r_i^m = \infty \forall i \in \{1, \dots, n\}, r_i^s = \infty \forall i \in \{1, \dots, n\}$

```

for  $i \leftarrow n$  down to 1 do
  if  $l \leq i$  then
     $t_i^s \leftarrow t_{i+1}^m, r_i^s \leftarrow v^T \cdot t_i^s + p_1, List.append((s, i, t_i^s, r_i^s)),$ 
     $t_i \leftarrow t_{i+1}^m + \frac{p_i - v^T \cdot t_{i+1}^m - p_1}{v^D},$ 
     $t_i^m \leftarrow \frac{p_i + v^D \cdot t_i - p_1}{v^D + v^T}, r_i^m \leftarrow v^T \cdot t_i^m + p_1,$ 
    for  $j \leftarrow l$  to  $i - 1$  do
      if  $p_j \leq v^T \cdot t_i + p_1$  and  $j \neq i - 1$  then
         $List.append\left(\left(j, \frac{p_j - p_1}{v^T}\right)\right)$ 
      else if  $p_j > v^T \cdot t_i + p_1$  then
         $List.append((v, i, t_i))$ 
        for  $h \leftarrow j$  to  $i - 1$  do
          if  $p_h \leq v^T \cdot t_i^m + p_1$  and  $h \neq i - 1$  then
             $List.append\left(\left(h, \frac{p_h - p_1}{v^T}\right)\right)$ 
          else if  $p_h > v^T \cdot t_i^m + p_1$  then
             $k \leftarrow h, List.append((m, i, t_i^m, r_i^m))$ 
            break
          else
             $k \leftarrow i, List.append\left(\left(i - 1, \frac{p_{i-1} - p_1}{v^T}\right)\right)$ 
            break
          end
        end
      end
    end
    break
  else
     $k \leftarrow i, List.append\left(\left(i - 1, \frac{p_{i-1} - p_1}{v^T}\right)\right)$ 
    break
  end
end
 $l \leftarrow k$ 
else
  break
end
end
return List

```

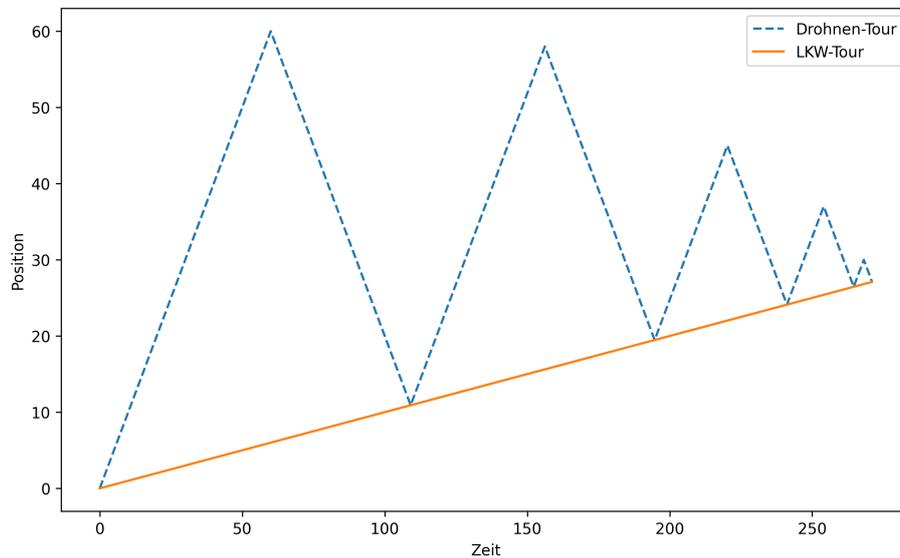
Beispiel 4.4.7. Seien $i^* = 1, v^D = 1$ und $v^T = 1/10$ gegeben. Es gibt keine Release Dates oder Deadlines. Die Positionen der Kunden sind der folgenden Tabelle zu entnehmen.

Kunde i	1	2	3	4	5	6	7	8
Position	0	10	20	30	37	45	58	60

Dann erhält man mit Algorithmus 4.4.6 die folgende optimale Lösung für die zu minimierende Zielfunktion $\max\{t_i, t_i^m\}$:

$$\begin{aligned} & \left((1, 0), (s, 8, 0, 0), (v, 8, 60, 60), (2, 100), (m, 8, 109.091, 10.909), \right. \\ & (s, 7, 109.091, 10.909), (v, 7, 156.182), (m, 7, 194.711, 19.471), \\ & (s, 6, 194.711, 19.471), (3, 200), (v, 6, 220.24), (m, 6, 241.127, 24.113), \\ & (s, 5, 241.127, 24.113), (v, 5, 254.014), (m, 5, 264.558, 26.456), \\ & \left. (s, 4, 264.558, 26.456), (v, 4, 268.103), (m, 4, 271.002, 27.1) \right) \end{aligned}$$

und damit erhält man die folgenden Touren:



Der Lösung ist zu entnehmen, dass die äußere For-Schleife in Algorithmus 4.4.6 erst für $i = 3$ abbricht. Es gilt nämlich

$$30 = p_4 > v^T \cdot t_5^m + x_0 = 1/10 \cdot 264.558$$

und

$$20 = p_3 \leq v^T \cdot t_4^m + x_0 = 1/10 \cdot 271.002 .$$

Lemma 4.4.8. *Algorithmus 4.4.6 löst das Problem der Minimierung von $\max\{t_i^m, t_i\}$ mit $i^* = p_1$. Die Laufzeitkomplexität ist $\mathcal{O}(n^2)$.*

Beweis. Der LKW fährt offensichtlich den Kunden entgegen, welche durch die Drohne bedient werden. Daher folgt aus 4.4.5 unmittelbar die Optimalität. Der Kunde mit der größten Entfernung, der noch nicht bedient wurde, wird durch die Drohne bedient. Zunächst wird geprüft, ob der LKW diesen Kunden bereits passiert hat, denn dann ist es nicht nötig, dass die Drohne ihn bedient. Die

Anzahl der Iterationen innerhalb der äußeren For-Schleife ist auf maximal n begrenzt. Innerhalb dieser wird erneut eine For-Schleife aufgerufen, die niemals mehr als n Iterationen durchläuft. Innerhalb dieser For-Schleife ist es möglich, dass erneut eine For-Schleife aufgerufen wird, deren Iterator beim aktuellen Wert des Iterators der zweiten For-Schleife startet und der Endwert des Iterators der dritten For-Schleife dem Endwert der zweiten For-Schleife entspricht. Außerdem wird nach dem Ende der dritten For-Schleife die zweite For-Schleife automatisch verlassen. Dadurch ergibt sich eine Laufzeitkomplexität des Algorithmus 4.4.6 von $\mathcal{O}(n^2)$. \square

Für das nächste Lemma wird die folgende Definition benötigt.

Definition 4.4.9. Gegeben sei eine feste LKW-Tour und eine feste Reihenfolge $\sigma = (\sigma(1), \dots, \sigma(n))$ von Kunden, welche von der Drohne bedient werden sollen. Der Zeitpunkt t_σ bezeichnet den Fertigstellungszeitpunkt des letzten Kunden, der von der Drohne bedient wird.

Lemma 4.4.10. Gegeben seien n Kunden mit $p_i < p_{i+1}$ und $k \in \{1, \dots, n\}$ mit $\sigma^* = (n, n-1, \dots, k-1, k+1, \dots, 2, 1, k)$, so dass $t_{\sigma^*} \leq t_\sigma$

für alle $\sigma \in \{(n, n-1, \dots, j+1, j-1, \dots, 2, 1, j) \mid j \in \{1, \dots, n\}\}$ gilt.

Zusätzlich gelte für die LKW-Tour $f^T(t) = v^T \cdot t + x_0$ die Abschätzung $f^T(t_{\sigma^*}) < p_1$. Dann ist σ^* ein optimaler Drohnen-Schedule bezüglich $\min \max\{t_i\}$ mit Ziel-funktionswert $z = t_{\sigma^*}$.

Beweis. Sei ein optimaler Drohnen-Schedule $(\sigma(1), \dots, \sigma(n))$ gegeben. Darüber hinaus sei $k = \sigma(n)$. Damit gilt $\{\sigma(1), \dots, \sigma(n-1)\} = \{1, 2, \dots, k-1, k+1, k+2, \dots, n-1, n\}$. Für jeden dieser $n-1$ Kunden muss die Drohne zurück zum LKW fliegen, also auch nachdem Kunde $\sigma(n-1)$ bedient wurde. Gemäß Lemma 4.4.5 existiert für das Problem $\min \max_i\{t_i^m, t_i\}$ mit den Kunden der Menge $\{1, 2, \dots, k-1, k+1, k+2, \dots, n-1, n\}$ ein optimaler Drohnen-Schedule $\tilde{\sigma} = (n, n-1, \dots, k+2, k+1, k-1, \dots, 2, 1)$. Eine solche Umsortierung der ersten $n-1$ Kunden verzögert den Begegnungszeitpunkt von Drohne und LKW nicht, nachdem die Kunden $\{\sigma(1), \dots, \sigma(n-1)\}$ bedient wurden. Es existiert daher ein optimaler Drohnen-Schedule mit der Struktur $(n, n-1, \dots, k+2, k+1, k-1, \dots, 2, 1, k)$. \square

Algorithmus 4.4.11 liefert einen optimalen Zielfunktionswert für dieses Problem. Dieser ist etwas weniger ausführlich, als Algorithmus 4.4.6, der im Gegensatz zu Algorithmus 4.4.11 auch eine optimale Lösung liefert.

Algorithmus 4.4.11

Data: Kunden $i \in \{1 = i^*, \dots, n\}$ mit $p_i < p_{i+1}$ und Geschwindigkeiten v^D und v^T

Result: Optimaler Zielfunktionswert für die Minimierung von $\max\{t_i\}$

Initialize: $z = \infty, k = -1, l = -1$

```

for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow i + 1$  to  $n$  do
    if  $t_{(n, n-1, \dots, j+1, j-1, \dots, i+2, i+1, j)} \leq z$  and  $\frac{p_i - p_1}{v^T} \leq z$  then
       $z \leftarrow \max \left\{ t_{(n, n-1, \dots, j+1, j-1, \dots, i+2, i+1, j)}, \frac{p_i - p_1}{v^T} \right\}$ 
       $k \leftarrow i$ 
       $l \leftarrow j$ 
    end
  end
end

```

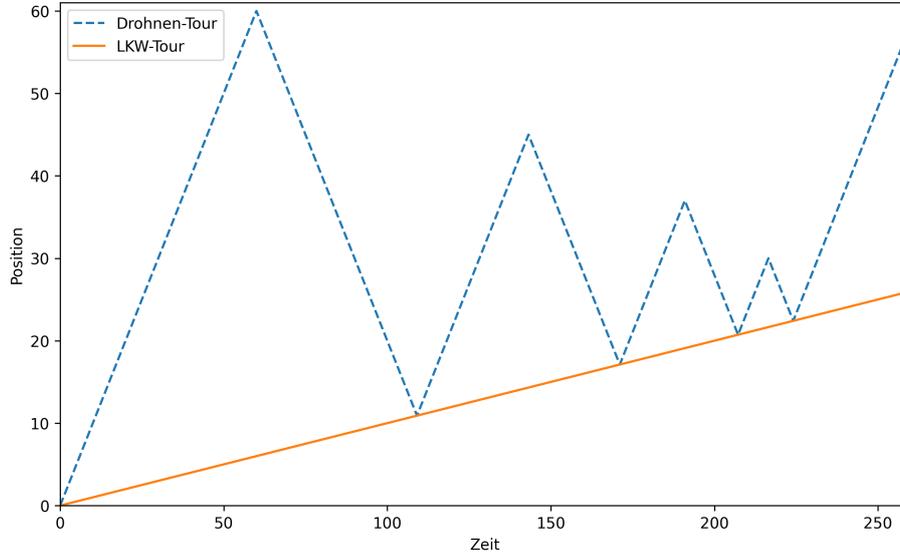
Lemma 4.4.12. *Algorithmus 4.4.11 löst das Problem der Minimierung von $\max\{t_i\}$ mit $i^* = 1$ optimal und hat eine Komplexität von $\mathcal{O}(n^2)$.*

Beweis. Da der LKW offensichtlich strikt den Kunden entgegenfährt, folgt mit 4.4.10 die Optimalität. Man erhält den LKW-Schedule $(1, \dots, k)$ und den Drohnen-Schedule $(n, n-1, \dots, l+1, l-1, \dots, 2, 1, l)$ mit Zielfunktionswert z . Die äußere For-Schleife umfasst maximal n Iterationen. Innerhalb dieser wird eine weitere For-Schleife aufgerufen, die nicht mehr als n Iterationen umfasst. Somit folgt die Komplexität von $\mathcal{O}(n^2)$ für den gesamten Algorithmus. \square

Beispiel 4.4.13. Gewählt wird die Instanz aus Beispiel 4.4.7, nun mit dem Ziel der Minimierung von $\max\{t_i\}$. Es ergibt sich die folgende optimale Lösung:

Kunde i	1	2	3	4	5	6	7	8
Position	0	10	20	30	37	45	58	60
t_i	0	100	200	216.518	190.967	143.182	259.697	60.0
r_i^m	-	-	-	22.411	20.724	17.107	-	10.909
t_i^m	-	-	-	224.108	207.243	171.074	-	109.091
r_i^s	-	-	-	20.724	17.107	10.909	22.411	0.0
t_i^s	-	-	-	207.243	171.074	109.091	224.108	0.0

Hierbei ist der Zielfunktionswert $t_7 = 259.697$. Auch hier werden die Kunden 2 und 3 durch den LKW bedient. Im Ort-Zeit-Diagramm ergeben sich die folgenden Touren:



Liegen also keine Zeitfenster vor und es gilt $i^* \in \{1, n\}$, so sind optimale Lösungen einfach zu konstruieren. Im Folgenden wird der Fall analysiert, wenn eine fest vorgegebene LKW-Tour vorliegt, wobei der LKW strikt in eine Richtung fährt und $i^* \notin \{1, n\}$ ist.

Lemma 4.4.14. Sei $v^D > v^T$ und eine LKW-Tour $f^T(t) = -v^T \cdot t + x_0$ gegeben. Darüber hinaus sind die Kunden $1, \dots, n$ mit $x_0 < p_i$ für alle $i \in \{1, \dots, n\}$ gegeben, die durch die Drohne mit dem Drohnen-Schedule $(\sigma(1), \dots, \sigma(n))$ ab Zeitpunkt $t_{\sigma(1)}^s$ bedient werden. Dann ist

$$t_{\sigma(n)} = t_{\sigma(1)}^s \left(1 + \frac{v^T}{v^D} \right) \left(\frac{v^D + v^T}{v^D - v^T} \right)^{n-1} + 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{p_{\sigma(n-k)} - x_0}{v^D} \right) \right) - \frac{p_{\sigma(n)}}{v^D} + \frac{x_0}{v^D}.$$

Beweis. Wie in 4.4.3. □

Analog gilt:

Lemma 4.4.15. Sei die gleiche Situation wie in Lemma 4.4.14 gegeben. Dann ist

$$t_n^m = t_{\sigma(1)}^s \left(\frac{v^D + v^T}{v^D - v^T} \right)^n + 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{p_{\sigma(n-k)} - x_0}{v^D - v^T} \right) \right).$$

Fährt der LKW strikt in Richtung der Drohnenkunden, so wählt die Drohne bezüglich $\min \max\{t_i^m, t_i\}$ den äußersten, noch nicht bedienten Kunden. Entfernt sich der LKW von den Kunden, so wählt die Drohne den nächsten noch nicht bedienten Kunden.

Lemma 4.4.16. Seien n Kunden gegeben mit $p_i > p_{i-1}$ sowie die LKW-Tour $f^T(t) = -v^T \cdot t + x_0$ mit $x_0 < p_1$. Dann ist der Drohnen-Schedule $(1, 2, \dots, n)$ für die gegebene LKW-Tour optimal bezüglich $\min \max\{t_i^m, t_i\}$.

Beweis. Die Drohne beginnt ab dem Zeitpunkt 0 damit, die Kunden zu bedienen. Der Begegnungszeitpunkt von Drohne und LKW, nachdem der letzte Kunde bedient wurde, ist gemäß Lemma 4.4.15 gleich

$$\begin{aligned} t_{\sigma(n)}^m &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{p_{\sigma(n-k)} - x_0}{v^D - v^T} \right) \right) \\ &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{-x_0}{v^D - v^T} \right) \right) \\ &\quad + \frac{2}{v^D - v^T} \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k p_{\sigma(n-k)} \right), \end{aligned}$$

wobei $\sigma(k)$ der k -te Kunde in einem gegebenen Schedule σ ist. Da $\left(\left(\frac{v^D + v^T}{v^D - v^T} \right)^k \right)_{k \in \mathbb{N}}$ eine streng monoton wachsende Folge ist und obiger Ausdruck minimiert werden soll, so ist die Zuordnung:

$$\sigma(k) = k$$

offenbar optimal. \square

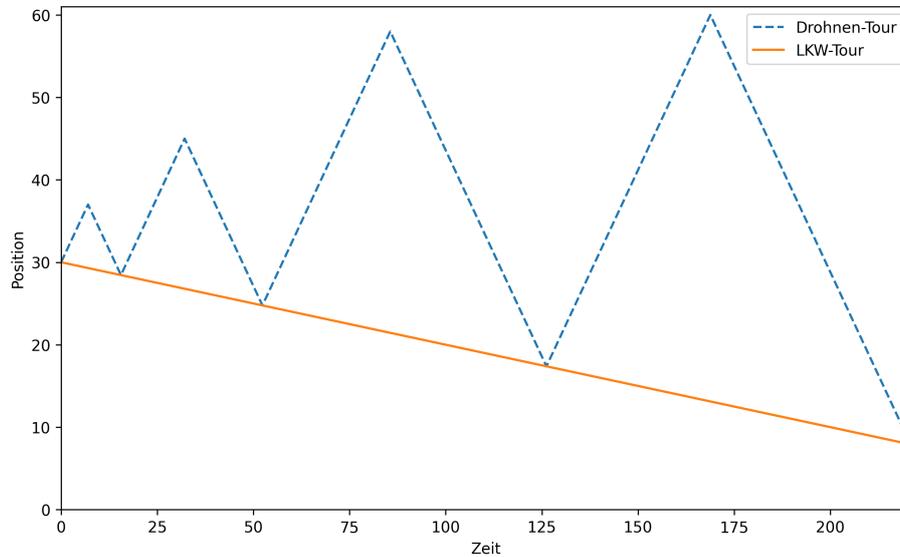
Lemma 4.4.17. *Sei die gleiche Situation wie in Lemma 4.4.16 gegeben. Dann ist der Drohnen-Schedule $(1, 2, \dots, n)$ für die gegebene LKW-Tour optimal bezüglich $\min \max\{t_i\}$.*

Beweis. Zunächst werden $n - 1$ Kunden gemäß Lemma 4.4.16 bedient. Dabei handelt es sich um die Kunden $1, \dots, n-1$. Der Kunde mit der größten Entfernung zum LKW wird als letzter Kunde durch die Drohne bedient. Ein Rückflug der Drohne ist von diesem Kunden nicht mehr notwendig. \square

Beispiel 4.4.18. Seien $x_0 = 30$, $v^D = 1$ und $v^T = 1/10$ gegeben. Ferner sei die LKW-Tour $f^T(t) = -v^T \cdot t + x_0$ vorgegeben sowie das Ziel die Minimierung von $\max\{t_i, t_i^m\}$. Dann erhält man mit 4.4.16 die folgende optimale Lösung mit dem Zielfunktionswert $z = t_4^m = 220.911$ für die angegebenen Positionen:

Kunde i	i^*	1	2	3	4
Position	30	37	45	58	60
t_i	0.0	7.0	32.111	85.58	168.82
r_i^m	-	28.444	24.765	17.38	7.909
t_i^m	-	15.556	52.346	126.2	220.911
r_i^s	-	30.0	28.444	24.765	17.38
t_i^s	-	0.0	15.556	52.346	126.2

Und man erhält somit die folgenden Touren:



Beispiel 4.4.19. Wird nun die Instanz aus obigem Beispiel 4.4.18 gewählt, jedoch mit dem Ziel der Minimierung von $\max\{t_i\}$, so erhält man nach 4.4.17 die gleiche Lösung wie in Beispiel 4.4.18, nur dass die Drohne nicht zum LKW zurückfliegen muss, nachdem sie Kunde 4 bedient hat. Daher ist der Zielfunktionswert $z = t_4 = 168.82$.

Anmerkung 4.4.20. Algorithmus 4.4.6 kann genutzt werden, um eine optimale Lösung sowohl für die Minimierung von $\max\{t_i\}$, als auch für die Minimierung von $\max\{t_i^m, t_i\}$ für eine gegebene LKW-Tour zu finden, bei der sich der LKW konstant von den Kunden entfernt. Hierbei ist die Geschwindigkeit des LKWs als negativ aufzufassen, und es muss nicht überprüft werden, ob der LKW einen Kunden bereits bedient hat.

Sei t^1 der Begegnungszeitpunkt von Drohne und LKW, nachdem der LKW von x_0 zu $v^T \cdot t^1 + x_0$ gefahren ist und die Drohne gemäß dem Drohnen-Schedule $(\sigma(1), \dots, \sigma(n))$ alle Kunden ohne Wartezeiten bedient hat. Dann ist anschaulich klar, dass der Begegnungszeitpunkt von Drohne und LKW ebenfalls t^1 ist, nachdem die Drohne alle Kunden gemäß dem Drohnen-Schedule $(\sigma(n), \dots, \sigma(1))$ bedient hat und die LKW-Tour

$$f^T(t) = -v^T \cdot t + (v^T \cdot t^1 + x_0)$$

lautet. Zur Illustration sei auf Abbildung 4.6 verwiesen.

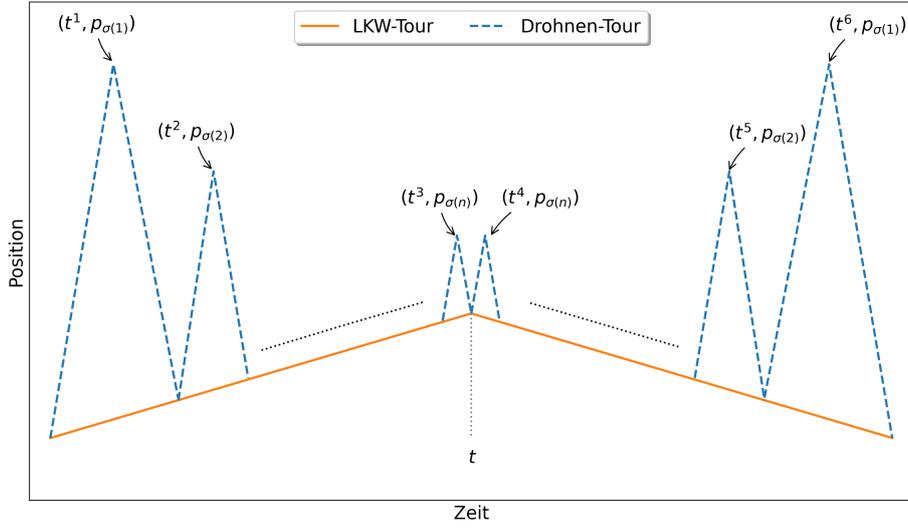


Abbildung 4.6: Drohnen-Schedule $(\sigma(1), \dots, \sigma(n))$ ab Zeitpunkt 0, wobei der LKW in die Richtung dieser Kunden fährt und Drohnen-Schedule $(\sigma(n), \dots, \sigma(1))$ ab Zeitpunkt t , wobei sich der LKW von diesen Kunden entfernt.

Im Folgenden wird dies formal beschrieben.

Lemma 4.4.21. Für $n \geq 1 \in \mathbb{N}$ gilt

$$1 + \left(\frac{2v^T}{v^D - v^T} \right) \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \right) = \left(\frac{v^D + v^T}{v^D - v^T} \right)^n.$$

Beweis. Die Behauptung lässt sich mittels der geometrischen Summenformel zeigen:

$$\begin{aligned} \sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k &= \frac{\left(\frac{v^D + v^T}{v^D - v^T} \right)^{(n-1)+1} - 1}{\left(\frac{v^D + v^T}{v^D - v^T} \right) - 1} = \frac{\left(\frac{v^D + v^T}{v^D - v^T} \right)^{(n-1)+1} - 1}{\left(\frac{v^D + v^T}{v^D - v^T} \right) - \left(\frac{v^D - v^T}{v^D - v^T} \right)} \\ &= \frac{\left(\frac{v^D + v^T}{v^D - v^T} \right)^{(n-1)+1} - 1}{\left(\frac{2v^T}{v^D - v^T} \right)} \\ \Leftrightarrow 1 + \left(\frac{2v^T}{v^D - v^T} \right) \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \right) &= \left(\frac{v^D + v^T}{v^D - v^T} \right)^n \quad \square \end{aligned}$$

Analog gilt die folgende Aussage:

Lemma 4.4.22. Für $n \geq 1 \in \mathbb{N}$ gilt

$$1 - \left(\frac{2v^T}{v^D + v^T} \right) \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \right) = \left(\frac{v^D - v^T}{v^D + v^T} \right)^n.$$

Lemma 4.4.23. Gegeben seien n Kunden mit den Positionen p_1, \dots, p_n sowie eine bijektive Abbildung $\sigma : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$. Des Weiteren sei

$$t := 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \left(\frac{p_{\sigma(n-k)} - x_0}{v^D + v^T} \right) \right).$$

Dann ist

$$t = 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{p_{\sigma(k+1)} - v^T \cdot t - x_0}{v^D - v^T} \right) \right).$$

Beweis. Es gilt

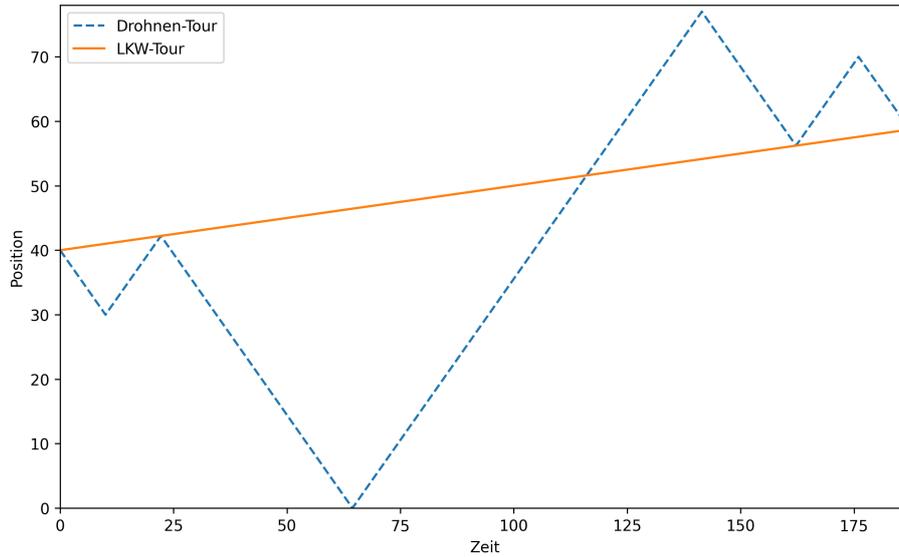
$$\begin{aligned} t &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \left(\frac{p_{\sigma(n-k)} - x_0}{v^D + v^T} \right) \right) \\ &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^{n-k-1} \left(\frac{p_{\sigma(k+1)} - x_0}{v^D + v^T} \right) \right) \\ &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^{n-k} \left(\frac{p_{\sigma(k+1)} - x_0}{v^D - v^T} \right) \right) \\ &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^{k-n} \left(\frac{p_{\sigma(k+1)} - x_0}{v^D - v^T} \right) \right). \end{aligned}$$

Somit lässt sich folgern:

$$\begin{aligned} t &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^{k-n} \left(\frac{p_{\sigma(k+1)} - x_0}{v^D - v^T} \right) \right) \\ \iff t \left(\frac{v^D + v^T}{v^D - v^T} \right)^n &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{p_{\sigma(k+1)} - x_0}{v^D - v^T} \right) \right) \\ \stackrel{4.4.21}{\iff} t \left(1 + \frac{2v^T}{v^D - v^T} \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \right) \right) &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{p_{\sigma(k+1)} - x_0}{v^D - v^T} \right) \right) \\ \iff t &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{p_{\sigma(k+1)} - x_0}{v^D - v^T} \right) \right) - \left(\frac{2v^T \cdot t}{v^D - v^T} \right) \cdot \sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \\ \iff t &= 2 \left(\sum_{k=0}^{n-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \left(\frac{p_{\sigma(k+1)} - v^T \cdot t - x_0}{v^D - v^T} \right) \right). \quad \square \end{aligned}$$

Unter der Voraussetzung, dass $p_i < p_{i+1}$ für alle $i \in \{1, \dots, n-1\}$ gilt und $i^* \notin \{1, n\}$ ist, sowie unter der Voraussetzung, dass der LKW strikt in eine Richtung fährt (o.B.d.A. Richtung p_n), lässt sich eine optimale Lösung mittels 4.4.10, 4.4.16 und 4.4.17 leicht konstruieren. Dies wird am folgenden Beispiel veranschaulicht:

Beispiel 4.4.24. Es sind fünf Kunden zu bedienen, wobei $i^* = 3$ mit $p_{i^*} = 40$ gilt. Der LKW nähert sich Kunde 5. Die Drohne hat eine Geschwindigkeit von 1, während der LKW eine Geschwindigkeit von $\frac{1}{10}$ hat. Die übrigen Parameter sind der Tabelle zu entnehmen. Die optimale Drohnen-Tour lässt sich wie folgt darstellen:



Kunde i	1	2	3	4	5	6	7
Position	0	30	40	50	55	70	77
t_i	64.444	10.0	0.0	100	150	176.0	141.444
r_i^m	51.605	42.222	-	-	-	58.727	56.222
t_i^m	116.049	22.222	-	-	-	187.273	162.222
r_i^s	42.222	40.0	-	-	-	56.222	51.605
t_i^s	22.222	0.0	-	-	-	162.222	116.049

Es werden also zunächst die Kunden in $\{1, \dots, i^*\}$ gemäß 4.4.17 bedient und anschließend die Kunden in $\{j, \dots, n\}$ mit $j \geq i^* + 1$ gemäß 4.4.5.

Unter der Voraussetzung, dass eine feste LKW-Tour gegeben ist, liegt ein Scheduling Problem vor, bei dem die Prozesszeiten vom Startzeitpunkt abhängen. Der Fertigstellungszeitpunkt und der Begegnungszeitpunkt hängen vom Startzeitpunkt der Drohne ab. Daher wird die folgende Definition eingeführt.

Definition 4.4.25. Im Falle einer festen LKW-Tour bezeichnet $t_i(t)$ den frühesten Zeitpunkt, zu dem die Drohne den Kunden p_i bedient, wenn die Drohne zum Zeitpunkt t vom LKW startet. Analog ist $t_i^m(t)$ der früheste Begegnungszeitpunkt, wenn die Drohne zum Zeitpunkt t vom LKW gestartet wird.

Im Folgenden wird der Fall betrachtet, in dem die Drohne eine feste Anzahl von Kunden bedienen soll, während der LKW eine feste Route hat, auf der er zunächst in Richtung dieser Kunden fährt und dann genau einmal wendet und sich nun entfernt. Die Voraussetzung, dass für den LKW eine feste LKW-Tour

gegeben ist, wird zwar nicht durch das hier vorliegende Modell abgebildet, jedoch wird dies später genutzt, da in einem nachfolgenden Komplexitätsbeweis eine LKW-Tour mit dieser Struktur erzwungen wird. Dies stellt also eine Vorbereitung für den Beweis von Satz 4.4.38 dar. Es wird nun stets die folgende LKW-Tour vorausgesetzt:

$$f^T = \begin{cases} x_0 + v^T \cdot t & \text{für } 0 \leq t \leq \frac{x_{turn} - x_0}{v^T} \\ -v^T \cdot t + 2 \cdot x_{turn} - x_0 & \text{für } t > \frac{x_{turn} - x_0}{v^T}. \end{cases} \quad (4.8)$$

Das bedeutet, dass der Lkw ab dem Zeitpunkt 0 zunächst konstant von x_0 bis x_{turn} fährt und bei x_{turn} wendet und konstant in die entgegengesetzte Richtung fährt.

Darüber hinaus gibt es die Kunden $1, \dots, n$ mit $p_i < p_{i+1}$ und $p_1 > x_{turn}$, die von der Drohne bedient werden. Für eine feste LKW-Tour ist das Problem ein Scheduling Problem mit startzeitabhängigen Prozesszeiten, die stückweise linear sind. Fährt der LKW zuerst auf die von der Drohne zu bedienenden Kunden zu, wendet genau einmal und entfernt sich dann von ihnen, so hängen die Zeitpunkte der Begegnung von Drohne und LKW vom Trennungszeitpunkt t' ab. Es wird zwischen den in (4.9) angegebenen Intervallen für den Trennungszeitpunkt unterschieden, die in Abbildung 4.7 visualisiert werden. Die Begegnungszeitpunkte von Drohne und LKW sind dann (4.10) zu entnehmen. Es sei an dieser Stelle darauf hingewiesen, dass das vorliegende Problem gewisse Ähnlichkeiten mit dem sogenannten 1-Maschinen-Scheduling-Problem mit v-förmigen Prozesszeiten aufweist, welches in (Sedding, 2020) Erwähnung findet. Eine detaillierte Betrachtung erfolgt zu einem späteren Zeitpunkt.

$$\begin{aligned} T_1(p_k) &= \left[0, \frac{v^D + v^T}{v^D - v^T} \cdot \left(\frac{x_{turn} - x_0}{v^T} - \frac{2p_k - 2x_0}{v^T + v^D} \right) \right] \\ T_2(p_k) &= \left(\frac{v^D + v^T}{v^D - v^T} \cdot \left(\frac{x_{turn} - x_0}{v^T} - \frac{2p_k - 2x_0}{v^T + v^D} \right), \frac{x_{turn} - x_0}{v^T} \right] \\ T_3(p_k) &= \left(\frac{x_{turn} - x_0}{v^T}, \infty \right) \end{aligned} \quad (4.9)$$

$$t_k^n(t') = \begin{cases} \frac{2p_k - 2x_0}{v^T + v^D} + t' \cdot \frac{v^D - v^T}{v^T + v^D} & \text{für } t' \in T_1(p_k) \\ \frac{2x_{turn} - 2p_k}{v^T - v^D} + t' & \text{für } t' \in T_2(p_k) \\ \frac{4x_{turn} - 2p_k - 2x_0}{v^T - v^D} - t' \cdot \frac{v^T + v^D}{v^T - v^D} & \text{für } t' \in T_3(p_k) \end{cases} \quad (4.10)$$

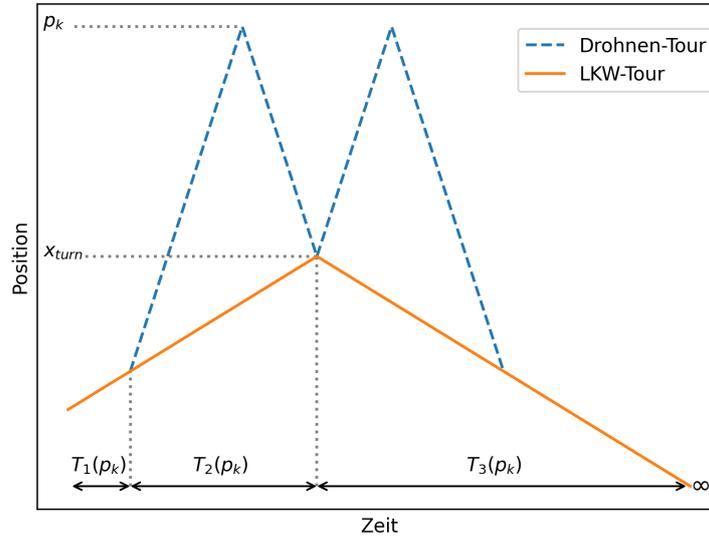


Abbildung 4.7: Intervalle der Trennungzeitpunkte

Zunächst wird geprüft, ob die oben definierte stückweise lineare Funktion tatsächlich die Begegnungszeitpunkte in Abhängigkeit vom Trennungzeitpunkt wiedergibt: Nach (4.7) ist offensichtlich, dass wenn der Begegnungszeitpunkt vor dem Wenden des LKWs stattfindet, so ist

$$t_k^m(t') = \frac{2p_k - 2x_0}{v^T + v^D} + t' \cdot \left(\frac{v^D - v^T}{v^T + v^D} \right).$$

Der Trennungzeitpunkt muss vor oder zum Zeitpunkt

$$\frac{v^D + v^T}{v^D - v^T} \cdot \left(\frac{x_{turn} - x_0}{v^T} - \frac{2p_k - 2x_0}{v^T + v^D} \right)$$

stattfinden. Dies gilt wegen der folgenden Äquivalenz

$$\begin{aligned} t &\leq \frac{v^D + v^T}{v^D - v^T} \left(\frac{x_{turn} - x_0}{v^T} - \frac{2p_k - 2x_0}{v^T + v^D} \right) \\ \Leftrightarrow \frac{2p_k - 2x_0}{v^T + v^D} + t \cdot \left(\frac{v^D - v^T}{v^T + v^D} \right) &\leq \frac{x_{turn} - x_0}{v^T}. \end{aligned}$$

Somit folgt

$$t_k^m(t') = \frac{2p_k - 2x_0}{v^T + v^D} + t' \cdot \left(\frac{v^D - v^T}{v^T + v^D} \right) \text{ für } t' \in T_1(p_k).$$

Sei nun $t' \in T_2(p_k)$, was bedeutet, dass die Drohne vor dem Wenden des LKWs startet, um einen Kunden zu bedienen, und dass sie nach dem Wenden des LKWs diesem begegnet. Auch hier wird nun wieder überprüft, ob (4.10) korrekt ist. Sei

$f_{(1,k,t')}^D$ die Drohnen-Tour ab dem Trennungszeitpunkt t' , bis zum Zeitpunkt t_k und f^{D_2} die Drohnen-Tour ab dem Fertigstellungszeitpunkt t_k , bis zum nächsten Begegnungszeitpunkt von Drohne und LKW. Dann ist

$$f_{(1,k,t')}^D(t) = v^D t + t'(v^T - v^D) + x_0 \text{ für } t \in [t', t_k],$$

mit

$$t_k = \frac{p_k - x_0 - t'(v^T - v^D)}{v^D}.$$

Somit ergibt sich

$$f_{(2,k,t')}^D(t) = -v^D t + 2p_k - x_0 - t'(v^T - v^D) \text{ für } t \in [t_k, t_k^m(t')],$$

denn der Schnittpunkt dieser Geraden mit der Geraden der LKW-Tour ergibt sich aus:

$$\begin{aligned} f_{(2,k,t')}^D(t_k^m) &= f^T(t_k^m) \\ \Leftrightarrow -v^D t_k^m + 2p_k - x_0 - t'(v^T - v^D) &= -v^T t_k^m + 2x_{turn} - x_0 \\ \Leftrightarrow t_k^m &= \frac{2p_k - 2x_{turn}}{v^D - v^T} + t' \end{aligned}$$

Folglich ist

$$t_k^m(t') = \frac{2p_k - 2x_{turn}}{v^D - v^T} + t' \text{ für } t \in T_2(p_k).$$

Nun wird das Intervall $T_3(p_k)$ aus (4.10) überprüft. Sei $t' \geq \frac{x_{turn} - x_0}{v^D}$ der Trennungszeitpunkt von Drohne und LKW, damit die Drohne Kunde k bedienen kann. Dann liegt die folgende Drohnen-Tour von diesem Zeitpunkt, bis zum Fertigstellungszeitpunkt t_k vor:

$$f_{(1,k,t')}^D(t) = v^D t + 2x_{turn} - x_0 - t'(v^D + v^T) \text{ für } t \in [t', t_k]$$

mit

$$t_k = \frac{p_k - 2x_{turn} + x_0 + t'(v^D + v^T)}{v^D}.$$

Fliegt nun die Drohne zum Zeitpunkt t_k zurück zum LKW, so ergibt sich die Drohnen-Tour:

$$f_{(2,k,t')}^D(t) = -v^D t + 2p_k - 2x_{turn} + x_0 + t'(v^D + v^T) \text{ für } t \in [t_k, t_k^m(t')],$$

denn es gilt:

$$\begin{aligned} f_{(2,k,t')}^D(t_k^m) &= f^T(t_k^m) \\ \Leftrightarrow -v^D t_k^m + 2p_k - 2x_{turn} + x_0 + t'(v^D + v^T) &= -v^T t_k^m + 2x_{turn} - x_0 \\ \Leftrightarrow t_k^m &= \frac{4x_{turn} - 2p_k - 2x_0}{v^T - v^D} - t' \left(\frac{v^T + v^D}{v^T - v^D} \right) \end{aligned}$$

Die Flugdauer der Drohne ab dem Trennungszeitpunkt bis zum nächsten Begegnungszeitpunkt ist als Funktion in Abbildung 4.8 dargestellt. Somit kann dieses Problem als ein 1-Maschinen-Scheduling Problem mit stückweise linearen

Prozesszeiten in Abhängigkeit des Zeitpunkts des Drohnenstarts zum Kunden aufgefasst werden.

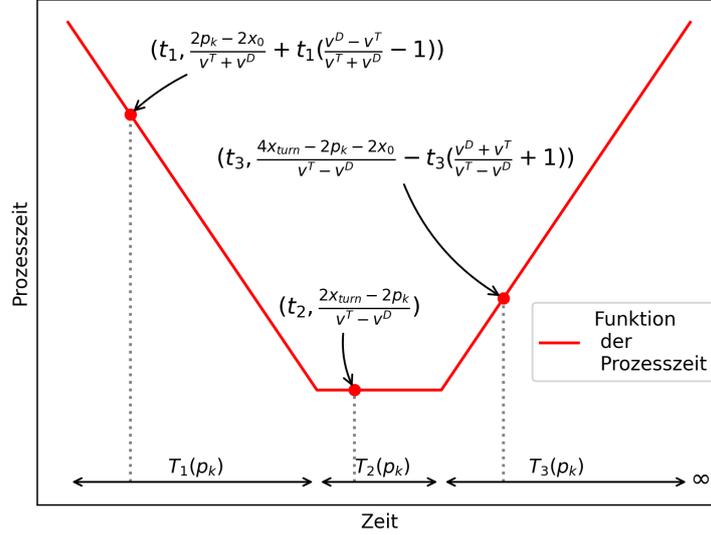


Abbildung 4.8: Differenz Trennungszeitpunkt und Begegnungszeitpunkt

Definition 4.4.26. Sei eine Instanz wie (4.8) gegeben. Ferner sei σ ein Drohnen-Schedule. Falls ein Kunde $\sigma(k)$ existiert, mit

$$t_{\sigma(k)}^s < \frac{x_{turn} - x_0}{v^T} \text{ und } t_{\sigma(k)}^m > \frac{x_{turn} - x_0}{v^T},$$

so wird Job $\sigma(k)$ als *Wendekunde* in σ bezeichnet.

Lemma 4.4.27. Sei eine Instanz wie in (4.8) und ein optimaler Drohnen-Schedule σ gegeben. Existiert ein Wendekunde $\sigma(k)$, so gelten

$$t_{\sigma(k)}^m \geq \frac{x_{turn} - x_0}{v^T} + \frac{x_{turn} - p_{\sigma(k)}}{v^T - v^D} \Rightarrow p_{\sigma(i)} \geq p_{\sigma(k)} \text{ für alle } i > k \quad (4.11)$$

$$\text{und } t_{\sigma(k)}^m < \frac{x_{turn} - x_0}{v^T} + \frac{x_{turn} - p_{\sigma(k)}}{v^T - v^D} \Rightarrow p_{\sigma(i)} \geq p_{\sigma(k)} \text{ für alle } i < k \quad (4.12)$$

Beweis. Sei ein optimaler Drohnen-Schedule σ gegeben, bei dem ein Wendekunde l existiert. Darüber hinaus sei $t := t_l^s$ bezüglich σ . Gezeigt wird hier nur die Implikation (4.11). Es wird nun also

$$t_l^m(t) \geq \frac{x_{turn} - x_0}{v^T} + \frac{x_{turn} - p_l}{v^T - v^D}. \quad (4.13)$$

vorausgesetzt. Das bedeutet, dass ein Teil des Hinflugs der Drohne zum Wendekunden l nach dem Wenden des LKW's stattfindet. Dies wird in Abbildung 4.9 dargestellt. In dem anderen Fall, d.h. (4.13) gilt nicht, erfolgt der Rückflug von Kunde l zum LKW, teilweise nachdem der LKW gewendet hat. Aus Symmetriegründen genügt es, die Behauptung nur für (4.13) zu zeigen, da der andere Fall analog ist. Sei r der Kunde, der nach Kunde l von der Drohne bedient wird.

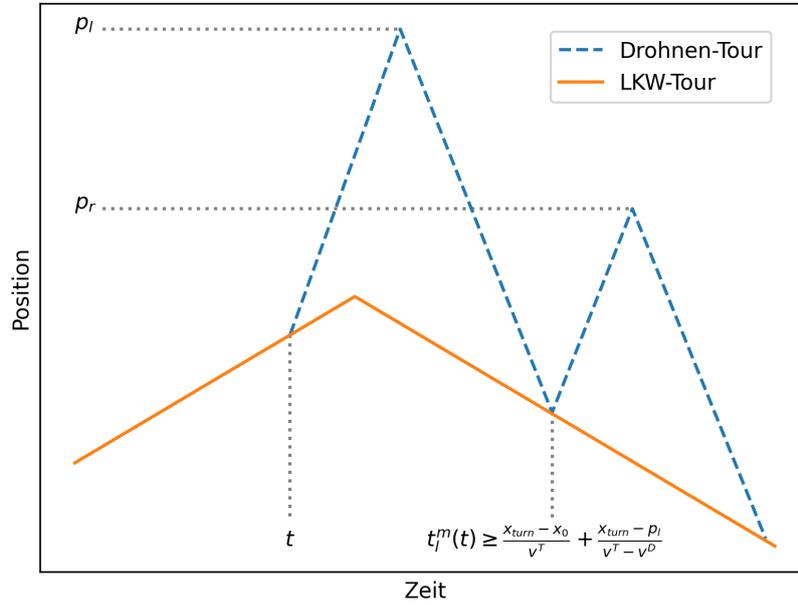


Abbildung 4.9: Hinflug der Drohne zum Wendekunden teilweise nach dem Wenden des LKWs

Angenommen es gilt $p_l > p_r$. Es werden die Begegnungszeitpunkte von r in σ und von r in σ nach Vertauschung der Kunden r und l untersucht. Hier müssen die Fälle $t \in T_1(p_r)$ bzw. $t \in T_2(p_r)$ unterschieden werden.

1. Fall: Ist $t \in T_1(p_r)$, so ist $t_r^m(t) \in T_2(p_l)$. Es wird nun gezeigt, dass die Ungleichung

$$t_r^m(t_l^m(t)) > t_l^m(t_r^m(t))$$

erfüllt ist. Es gilt

$$\begin{aligned}
 & p_l v^T + p_r v^D > p_r v^T + x_{turn} v^D \\
 \Leftrightarrow & p_l v^D - x_{turn} v^D > p_l v^D - p_l v^T - p_r v^D + p_r v^T \\
 \Leftrightarrow & (p_l - x_{turn}) v^D > (p_l - p_r)(v^D - v^T) \\
 \Leftrightarrow & \frac{p_l - x_{turn}}{v^D - v^T} > \frac{p_l - p_r}{v^D} \tag{4.14}
 \end{aligned}$$

Nun wird $t_l^m(t)$ abgeschätzt, um damit später obige Ungleichung zu zeigen:

$$\begin{aligned}
 t_l^m(t) & \stackrel{(4.13)}{\geq} \frac{x_{turn} - x_0}{v^T} + \frac{p_l - x_{turn}}{v^D - v^T} \stackrel{(4.14)}{>} \frac{x_{turn} - x_0}{v^T} + \frac{p_l - p_r}{v^D} \\
 & = \frac{4x_{turn}v^D - 4x_0v^D + 4p_l v^T - 4p_r v^T}{4v^T v^D} \\
 & = \frac{2x_{turn}v^D + 2p_l v^D - 2p_r v^D - 2x_0v^D + 2x_{turn}v^T + 2p_l v^T}{4v^T v^D} \\
 & \quad + \frac{-2p_r v^T - 2x_0v^T + 2x_{turn}v^D - 2x_{turn}v^T - 2p_l v^D + 2p_l v^T}{4v^T v^D}
 \end{aligned}$$

$$\begin{aligned}
& + \frac{+2p_r v^D - 2p_r v^T - 2x_0 v^D + 2x_0 v^T}{4v^T v^D} \\
& = \frac{(2x_{turn} + 2p_l - 2p_r - 2x_0)(v^D + v^T)}{4v^T v^D} \\
& + \frac{(2x_{turn} - 2p_l)(v^D - v^T) + (2p_r - 2x_0)(v^D - v^T)}{4v^T v^D} \\
& = \frac{(-2x_{turn} - 2p_l + 2p_r + 2x_0)(v^D - v^T)(v^D + v^T)}{(v^T - v^D)4v^T v^D} \\
& - \frac{(2x_{turn} - 2p_l)(v^D - v^T)^2(v^D + v^T)}{(v^T - v^D)(v^D + v^T)4v^D v^T} \\
& + \frac{(2p_r - 2x_0)(v^D - v^T)(v^D + v^T)}{(v^T + v^D)4v^T v^D}
\end{aligned}$$

Außerdem ist

$$\begin{aligned}
\frac{4v^D v^T}{(v^D - v^T)(v^D + v^T)} & = \frac{(v^{D^2} + 2v^D v^T + v^{T^2}) - (v^{D^2} - 2v^D v^T + v^{T^2})}{(v^D - v^T)(v^D + v^T)} \\
& = \frac{(v^D + v^T)^2 - (v^D - v^T)^2}{(v^D - v^T)(v^D + v^T)} = \frac{v^D + v^T}{v^D - v^T} - \frac{v^D - v^T}{v^D + v^T}.
\end{aligned}$$

Daher ist

$$\begin{aligned}
t_l^m(t) \left(\frac{v^D + v^T}{v^D - v^T} - \frac{v^D - v^T}{v^D + v^T} \right) & > \frac{-2x_{turn} - 2p_l + 2p_r + 2x_0}{v^T - v^D} \\
& - \frac{(2x_{turn} - 2p_l)(v^D - v^T)}{(v^T - v^D)(v^D + v^T)} + \frac{2p_r - 2x_0}{v^T + v^D} \\
\iff \frac{4x_{turn} - 2p_r - 2x_0}{v^T - v^D} - t_l^m(t) \left(\frac{v^D + v^T}{v^T - v^D} \right) & > \frac{2x_{turn} - 2p_l}{v^T - v^D} + \frac{2p_r - 2x_0}{v^T + v^D} \\
& + \left(t_l^m(t) - \frac{2x_{turn} - 2p_l}{v^T - v^D} \right) \left(\frac{v^D - v^T}{v^D + v^T} \right) \\
\iff t_r^m(t_l^m(t)) > t_l^m \left(t_r^m \left(t_l^m(t) - \frac{2x_{turn} - 2p_l}{v^T - v^D} \right) \right) & = t_l^m(t_r^m(t))
\end{aligned}$$

Die letzte Gleichung ist erfüllt, da $t_l^m(t) = \frac{2x_{turn} - 2p_l}{v^T - v^D} + t$ gilt. Somit kann σ nicht optimal gewesen sein.

2.Fall: Ist $t \in T_2(p_r)$, so ist $t_r^m(t) \in T_3(p_l)$. Somit gilt mit (4.10)

$$\begin{aligned}
& t_r^m(t_l^m(t)) - t_l^m(t_r^m(t)) \\
& = \frac{4x_{turn} - 2p_r - 2x_0}{v^T - v^D} - \left(\frac{2x_{turn} - 2p_l}{v^T - v^D} + t \right) \left(\frac{v^T + v^D}{v^T - v^D} \right) \\
& - \left(\frac{4x_{turn} - 2p_l - 2x_0}{v^T - v^D} - \left(\frac{2x_{turn} - 2p_r}{v^T - v^D} + t \right) \left(\frac{v^T + v^D}{v^T - v^D} \right) \right) \\
& = \frac{2p_l - 2p_r}{v^T - v^D} - \frac{2p_l - 2p_r}{v^T - v^D} \left(\frac{v^T + v^D}{v^D - v^T} \right)
\end{aligned}$$

$$= \frac{2p_l - 2p_r}{v^D - v^T} \left(\frac{v^T + v^D}{v^D - v^T} - 1 \right) > 0. \quad \square$$

Definition 4.4.28. Sei eine Instanz wie in (4.8) und ein Drohnen-Schedule σ gegeben. Die Mengen L_σ und R_σ sind definiert als:

$$L_\sigma := \left\{ i \in \{1, \dots, n\} : t_i^m \leq \frac{x_{turn} - x_0}{v^T} + \frac{x_{turn} - p_i}{v^T - v^D} \right\}$$

und

$$R_\sigma := \left\{ i \in \{1, \dots, n\} : t_i^s > \frac{x_{turn} - x_0}{v^T} - \frac{x_{turn} - p_i}{v^T - v^D} \right\}.$$

Nach Lemma 4.4.27 gilt offensichtlich, dass für einen optimalen Drohnen-Schedule σ der Wendekunde l , entweder in L_σ liegt und $p_l \leq p_i$ für alle $i \in L_\sigma$ erfüllt ist, oder in R_σ liegt und $p_l \leq p_i$ für alle $i \in R_\sigma$ ist.

Definition 4.4.29. Gegeben seien eine LKW-Tour $f^T : \mathbb{R} \rightarrow \mathbb{R}$ (stückweise linear und stetig) und ein Kunde k , der durch die Drohne bedient wird. Dann bezeichnet $t^m(k, t)$ mit $t \in \mathbb{R}$ und $k \in \{1, \dots, n\}$ den Begegnungszeitpunkt von Drohne und LKW, nachdem die Drohne zum Zeitpunkt t vom LKW startet, Kunde k bedient und sich anschließend sofort zum LKW bewegt.

Definition 4.4.30. Seien $k \geq 0$ Kunden der Menge $\{1, \dots, k\}$ mit $p_i < p_{i+1}$ für alle $i \in \{1, \dots, k-1\}$ gegeben. Dann ist

$$V(t, \{1, \dots, k\}) := \begin{cases} t & \text{falls } k = 0 \\ \min \left\{ V(t^m(k, t), \{1, \dots, k-1\}), t^m(k, V(t, \{1, \dots, k-1\})) \right\} & \text{sonst} \end{cases}.$$

Hierbei ist für $l = 0$ die Menge $\{1, \dots, l\}$ gleich der leeren Menge.

Lemma 4.4.31. Sei die eine Instanz wie in (4.8) gegeben. Dann ist der frühest mögliche Begegnungszeitpunkt von LKW und Drohne gleich $V(0, \{1, \dots, n\})$, nachdem die Drohne alle Kunden bedient hat.

Beweis. Existiert nur ein Kunde, so ist

$$V(0, \{1\}) = \min\{t^m(1, 0), t^m(1, V(0, \emptyset))\} = \min\{t^m(1, 0), t^m(1, 0)\} = t^m(1, 0).$$

Es sei angenommen, dass der früheste Begegnungszeitpunkt von Drohne und LKW gleich $V(t^m(n, 0), \{1, \dots, n-1\})$ ist, nachdem die Drohne ab dem Zeitpunkt $t^m(n, 0)$ die Kunden $1, \dots, n-1$ bedient hat. Der zugehörige optimale Drohnen-Schedule der Kunden $1, \dots, n-1$ wird mit σ_1 bezeichnet. Analog sei angenommen, dass der früheste Begegnungszeitpunkt von Drohne und LKW gleich $V(0, \{1, \dots, n-1\})$ ist, nachdem die Drohne ab dem Zeitpunkt 0 die Kunden $1, \dots, n-1$ bedient hat. Auch hier wird mit σ_2 der zugehörige optimale Drohnen-Schedule der Kunden $1, \dots, n-1$ bezeichnet. Im Folgenden wird unterschieden, ob der Flug der Drohne, vom LKW zu Kunde n und zurück, vor oder nach dem Wenden des LKWs stattfindet. Falls der Flug der Drohne vor dem Wenden stattfindet, wird Kunde n in einer optimalen Lösung unter dieser Voraussetzung gemäß Lemma 4.4.5 zum Zeitpunkt 0 bedient. Danach werden ab

dem Zeitpunkt $t^m(n, 0)$ die Kunden des Schedules σ_1 bedient. Findet der Flug der Drohne nach dem Wenden statt, wird gemäß Lemma 4.4.16 Kunde n als letzter Kunde bedient, d.h. zunächst ab dem Zeitpunkt 0 die Kunden des Schedules σ_2 und anschließend ab dem Zeitpunkt $V(0, \{1, \dots, n-1\})$ der Kunde n . Dieser wäre lediglich ein Wendekunde, sofern die Drohnen-Tour ab Zeitpunkt 0 zu Kunde n und zurück zum LKW nach dem Wenden des LKWs beendet wäre, oder falls die Kunden $1, \dots, n-1$ alle vor dem Wenden des LKWs bedient wurden und die Drohne dem LKW vor dem Wenden begegnet. Unter den Optionen, ob die Drohnen-Tour bezüglich Kunde n vor, nach oder während des Wendens stattfindet, wird diejenige mit dem geringsten Zielfunktionswert gewählt. \square

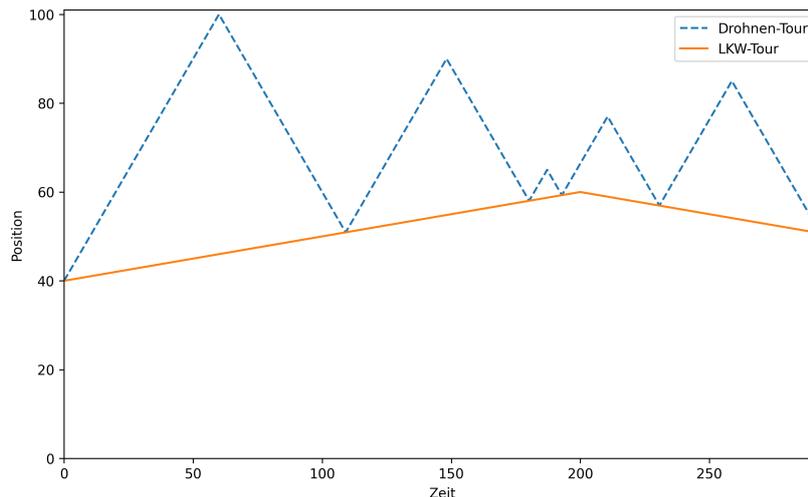
Beispiel 4.4.32. Gegeben sei eine festgelegte LKW-Tour

$$f^T = \begin{cases} 40 + \frac{1}{10} \cdot t & \text{für } 0 \leq t \leq 200 \\ -\frac{1}{10} \cdot t + 80 & \text{für } t > 200 \end{cases},$$

wobei also $x_0 = 40, x_{turn} = 60$ und $t_{turn} = 200$ ist. Die Drohne hat eine Geschwindigkeit von $v^D = 1$. Es ergibt sich die folgende optimale Lösung für die Minimierung von $\max\{t_i^m, t_i\}$:

Kunde i	1	2	3	4	5
Position	65	77	85	90	100
t_i	187.149	210.576	258.704	148.182	60.0
r_i^m	59.286	56.936	50.7	58.017	50.909
t_i^m	192.863	230.64	293.005	180.165	109.091
r_i^s	58.017	59.286	56.936	50.909	40.0
t_i^s	180.165	192.863	230.64	109.091	0.0

Es ergeben sich die folgenden Touren:



Das Optimum lässt sich mit 4.4.31 aus den folgenden Teilproblemen ableiten.

$$\begin{aligned} V(0, \{1, 2, 3, 4, 5\}) &= V(t^m(5, 0), \{1, 2, 3, 4\}) = V(109.091, \{1, 2, 3, 4\}) \\ &= V(t^m(4, 109.091), \{1, 2, 3\}) = V(180.165, \{1, 2, 3\}) \\ &= t^m(3, V(180.165, \{1, 2\})). \end{aligned}$$

Hierbei ist $V(180.165, \{1, 2\}) = t^m(2, V(180.165, \{1\}))$ und es gilt

$$V(180.165, \{1\}) = t^m(1, 180.165) = 192.836.$$

Damit ergibt sich die Gleichung $V(180.165, \{1, 2\}) = t^m(2, 192.836)$ und insgesamt folgt

$$V(0, \{1, 2, 3, 4, 5\}) = t^m(3, V(180.165, \{1, 2\})) = t^m(3, 230.64) = 293.005.$$

Der Wendekunde ist Kunde 2 und es gilt

$$t_2^s = t_1^m = 192.863 > 181.1 = \frac{x_{turn} - x_0}{v^T} - \frac{x_{turn} - p_2}{v^T - v^D}.$$

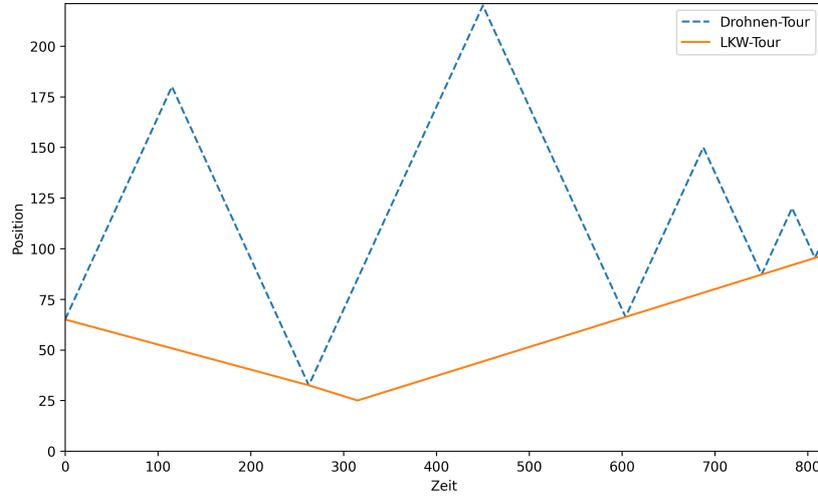
Daher liegt nach Definition 4.4.28 die folgende Partitionierung der Kunden vor:

$$L_\sigma = \{1, 4, 5\} \text{ und } R_\sigma = \{2, 3\}.$$

Im Falle einer LKW-Tour, bei der sich der LKW zunächst entfernt, wendet und strikt den Kunden entgegenfährt, lässt sich durch Anwenden von Lemma 4.4.31 eine optimale Lösung erzielen. Allerdings wird unter einer betrachteten Drohnenkundenmenge der Kunde mit der geringsten Entfernung als erstes oder letztes durch die Drohne bedient. Im Folgenden wird für das Beispiel $v^D = 1$ angenommen.

$$f^T = \begin{cases} 65 - \frac{1}{7} \cdot t & \text{für } 0 \leq t \leq 315 \\ \frac{1}{7} \cdot t - 25 & \text{für } t > 315 \end{cases}$$

Kunde i	1	2	3	4	5	6
Position	65	100	120	150	180	220
t_i	0.0	812.3	783.1	687.5	115.0	450.0
r_i^m	-	96.5	95.4	87.2	32.5	66.2
t_i^m	-	815.8	807.7	750.3	262.5	603.8
r_i^s	-	95.4	87.2	66.2	65.0	32.5
t_i^s	-	807.7	750.3	603.8	0.0	262.5



Im Folgenden wird das Line-TSPwD ohne Zeitfenster mit $i^* \notin \{1, n\}$ analysiert. Es wird zwar keine feste LKW-Tour vorausgesetzt, jedoch wird eine LKW-Tour wie (4.8) durch die im Beweis von Satz 4.4.38 angegebene Reduktion erzwungen. Wie bereits erwähnt, ähnelt dieses Problem dem Scheduling Problem mit zeitabhängigen Bearbeitungszeiten mit einer v-förmigen Bestrafungsfunktion ϖ_j für jeden Job j , wie in (Sedding, 2020) beschrieben. Genauer gesagt beinhaltet dieses Problem Jobs j , deren Prozesszeiten wie folgt definiert sind:

$$p_j = l_j + \varpi_j(t)$$

mit $l_j > 0$ (Basisprozesszeit) und

$$\begin{aligned} \varpi_j : \mathbb{R}_{\geq 0} &\rightarrow \mathbb{R}_{\geq 0} \\ t &\mapsto \max\{-a_j(t - \tau), b_j(t - \tau)\}, \end{aligned}$$

wobei $a_j, b_j, \tau \geq 0$ und $a_j < 1$.

Mithilfe des Even-Odd-Partition-Problems wurde in (Sedding, 2020) gezeigt, dass das Problem mit $a_j = a$ und $b_j = b$ für alle Jobs j \mathcal{NP} -schwer ist.

Definition 4.4.33. (Garey, Tarjan & Wilfong, 1988) Seien $n \in \mathbb{Z}_{>0}$ und $X = \{x_1, \dots, x_{2n}\}$ mit $x_1, \dots, x_{2n} \in \mathbb{Z}_{>0}$ und $x_i < x_{i+1}$ für alle $i \in \{1, \dots, 2n-1\}$ gegeben. Dann heißt das folgende Entscheidungsproblem *Even-Odd-Partition-Problem*: „Gibt es eine Partitionierung X_1 und $X_2 = X \setminus X_1$, sodass für alle $i \in \{1, \dots, n\}$ entweder x_{2i} oder x_{2i-1} in X_1 liegt und die Gleichung $\sum_{x \in X_1} x = \sum_{x \in X_2} x$ erfüllt ist?“

Lemma 4.4.34. ((Garey et al., 1988), Lemma 1) Das Even-Odd-Partition-Problem ist \mathcal{NP} -vollständig.

Ähnlich, aber in angepasster Form zu (Sedding, 2020), wird eine Reduktion des Even-Odd-Partitions-Problems auf das Line-TspwD konstruiert. Hier wird eine feste LKW-Tour wie in 4.4.31 erzwungen. Dies kann realisiert werden, indem zwei Kunden nur durch den LKW bedient werden können und die Reisedauer des LKWs, um beide zu bedienen, genau dem Schwellenwert des Entscheidungsproblems entspricht. Die Werte x_i aus einer Even-Odd-Partition-Problem-Instanz

werden zu Drohnenkunden transformiert. Es wird sich zeigen, dass die Flugdauer, um einen solchen Kunden anzufliegen und zurück zum LKW, genau x_i beträgt, falls eine zulässige Lösung vorliegt. Die Mengen X_1 und X_2 werden den Mengen L_σ und R_σ entsprechen, wobei kein Wendekunde existiert, falls eine zulässige Lösung vorliegt. Diese Zusammenhänge werden später im Beweis von Satz 4.4.38 im Detail gezeigt. Wird diese Idee auf eine Reduktion des 1-Maschinen-Scheduling-Problems mit zeitabhängigen v-förmigen Prozesszeiten auf das Line-TSPwD angewendet, so ergeben sich Probleme bei Aufträgen mit einer Basisprozesszeit von 0, da zum Wendzeitpunkt genau diese Kunden vom LKW bedient werden. Aufträge mit einer Basisprozesszeit von 0 spielen bei der Reduktion des Even-Odd-Partition-Problems auf das 1-Maschinen-Scheduling-Problem mit zeitabhängigen v-förmigen Prozesszeiten eine wichtige Rolle, die jedoch in dieser Form nicht auf das Line-TSPwD übertragbar ist.

Ferner ist für einen Wendekunden beim Line-TSPwD zu beachten, dass dessen Prozesszeit für einen gewissen Zeitraum konstant ist (siehe Abbildung 4.8 Intervall T_2). Dies ist beim 1-Maschinen Scheduling Problem mit v-förmigen Prozesszeiten nicht der Fall. Zunächst werden einige Aussagen bewiesen, die im Beweis von 4.4.38 benötigt werden.

Definition 4.4.35. Seien eine Line-TSPwD-Instanz, eine feste LKW-Tour und ein Drohnen-Schedule σ gegeben. Dann bezeichnet $t^m(\sigma, t)$ den Begegnungszeitpunkt von Drohne und LKW, nachdem alle Drohnenkunden ab Zeitpunkt t gemäß σ durch die Drohne bedient wurden.

Lemma 4.4.36. Gegeben sei eine LKW-Tour wie in 4.8 sowie zwei Kunden i und j mit den Positionen $x_{turn} < p_i < p_j$. Ist $t \in T_2(p_i)$, dann ist $t \in T_2(p_j)$.

Beweis. Es gilt

$$\begin{aligned} t \in T_2(p_i) &= \left(\left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{x_{turn} - x_0}{v^T} - \frac{2p_i - 2x_0}{v^T + v^D} \right), \frac{x_{turn} - x_0}{v^T} \right] \\ &\subset \left(\left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{x_{turn} - x_0}{v^T} - \frac{2p_j - 2x_0}{v^T + v^D} \right), \frac{x_{turn} - x_0}{v^T} \right] \\ &= T_2(p_j). \quad \square \end{aligned}$$

Lemma 4.4.37. Seien eine LKW-Tour wie in 4.8 und n Kunden mit $x_{turn} < p_1$ und $p_i < p_{i+1}$ für alle $i \in \{1, \dots, n-1\}$ gegeben und es gelte

$$p_n > \frac{(x_{turn} - x_0)(v^D - v^T)}{v^T} + x_{turn}. \quad (4.15)$$

Dann wird Kunde n in einer optimalen Lösung als letzter Kunde bedient.

Beweis. Sei σ ein optimaler Schedule. Angenommen, $n = \sigma(k) \neq \sigma(n)$. Ferner seien $\bar{\sigma} := \sigma|_{\{1, \dots, k-1\}}$ die Einschränkung von σ auf $\{1, \dots, k-1\}$ und $t := t^m(\bar{\sigma}, 0)$. Entweder ist $t \in T_2(p_{\sigma(k)})$ oder $t \in T_3(p_{\sigma(k)})$, da

$$\begin{aligned}
& \left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{x_{turn} - x_0}{v^T} - \frac{2p_{\sigma(k)} - 2x_0}{v^T + v^D} \right) \\
& < \left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{x_{turn} - x_0}{v^T} - \frac{2 \left(\frac{(x_{turn} - x_0)(v^D - v^T)}{v^T} + x_{turn} \right) - 2x_0}{v^T + v^D} \right) \\
& = \left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{x_{turn} - x_0}{v^T} - \left(\frac{v^D - v^T}{v^D + v^T} \right) \frac{2x_{turn} - 2x_0}{v^T} + \frac{2x_{turn} - 2x_0}{v^D + v^T} \right) \\
& = \left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{(x_{turn} - x_0)(v^D + v^T)}{(v^D + v^T)v^T} \right. \\
& \quad \left. - \frac{(2x_{turn} - 2x_0)(v^D - v^T) + (2x_{turn} - 2x_0)v^T}{(v^D + v^T)v^T} \right) \\
& = \left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{x_0v^D - x_0v^T - x_{turn}v^D + x_{turn}v^T}{(v^D + v^T)v^T} \right) \\
& = \left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{(v^D - v^T)(x_0 - x_{turn})}{(v^D + v^T)v^T} \right) = \frac{x_0 - x_{turn}}{v^T} < 0
\end{aligned}$$

Außerdem gilt $t_{\sigma(k)}^m(t) \in T_3(p_{\sigma(k+1)})$. Im Folgenden wird untersucht, wie sich der Zielfunktionswert ändert, wenn $\sigma(k)$ und $\sigma(k+1)$ im Schedule σ getauscht werden.

1. Fall : $t \in T_2(p_{\sigma(k+1)}) \stackrel{4.4.36}{\Rightarrow} t \in T_2(p_{\sigma(k)})$

$$\begin{aligned}
& t_{\sigma(k+1)}^m \left(t_{\sigma(k)}^m(t) \right) - t_{\sigma(k)}^m \left(t_{\sigma(k+1)}^m(t) \right) \\
& = \frac{2p_{\sigma(k+1)} + 2x_0 - 4x_{turn}}{v^D - v^T} + \left(\frac{2p_{\sigma(k)} - 2x_{turn}}{v^D - v^T} + t \right) \left(\frac{v^T + v^D}{v^D - v^T} \right) \\
& \quad - \left(\frac{2p_{\sigma(k)} + 2x_0 - 4x_{turn}}{v^D - v^T} + \left(\frac{2p_{\sigma(k+1)} - 2x_{turn}}{v^D - v^T} + t \right) \left(\frac{v^D + v^T}{v^D - v^T} \right) \right) \\
& = \left(\frac{2p_{\sigma(k+1)} - 2x_{turn}}{v^D - v^T} \right) \left(1 - \frac{v^D + v^T}{v^D - v^T} \right) + \frac{2x_0 - 2x_{turn}}{v^D - v^T} \\
& \quad + \left(\frac{2p_{\sigma(k)} - 2x_{turn}}{v^D - v^T} \right) \left(\frac{v^D + v^T}{v^D - v^T} - 1 \right) - \frac{2x_0 - 2x_{turn}}{v^D - v^T} \\
& = \left(\frac{2x_{turn} - 2p_{\sigma(k+1)} + 2p_{\sigma(k)} - 2x_{turn}}{v^D - v^T} \right) \left(\frac{v^D + v^T}{v^D - v^T} - 1 \right) \\
& = \left(\frac{2p_{\sigma(k)} - 2p_{\sigma(k+1)}}{v^D - v^T} \right) \left(\frac{v^D + v^T}{v^D - v^T} - 1 \right) > 0
\end{aligned}$$

2. Fall : $t \in T_3(p_{\sigma(k+1)}) (\Rightarrow t \in T_3(p_{\sigma(k)}))$

$$\begin{aligned}
& t_{\sigma(k+1)}^m \left(t_{\sigma(k)}^m(t) \right) - t_{\sigma(k)}^m \left(t_{\sigma(k+1)}^m(t) \right) \\
& = t_{\sigma(k+1)}^m \left(\frac{2p_{\sigma(k)} + 2x_0 - 4x_{turn}}{v^D - v^T} + t \left(\frac{v^D + v^T}{v^D - v^T} \right) \right)
\end{aligned}$$

$$\begin{aligned}
& -t_{\sigma(k)}^m \left(\frac{2p_{\sigma(k+1)} + 2x_0 - 4x_{turn}}{v^D - v^T} + t \left(\frac{v^D + v^T}{v^D - v^T} \right) \right) \\
&= \frac{2p_{\sigma(k+1)} + 2x_0 - 4x_{turn}}{v^D - v^T} + \left(\frac{2p_{\sigma(k)} + 2x_0 - 4x_{turn}}{v^D - v^T} \right. \\
&\quad \left. + t \left(\frac{v^D + v^T}{v^D - v^T} \right) \right) \left(\frac{v^D + v^T}{v^D - v^T} \right) \\
&\quad - \frac{2p_{\sigma(k)} + 2x_0 - 4x_{turn}}{v^D - v^T} - \left(\frac{2p_{\sigma(k+1)} + 2x_0 - 4x_{turn}}{v^D - v^T} \right. \\
&\quad \left. + t \left(\frac{v^D + v^T}{v^D - v^T} \right) \right) \left(\frac{v^D + v^T}{v^D - v^T} \right) \\
&= \frac{2p_{\sigma(k+1)} - 2p_{\sigma(k)}}{v^D - v^T} + \frac{2p_{\sigma(k)} - 2p_{\sigma(k+1)}}{v^D - v^T} \left(\frac{v^D + v^T}{v^D - v^T} \right) \\
&= \frac{2p_{\sigma(k)} - 2p_{\sigma(k+1)}}{v^D - v^T} \left(\frac{v^D + v^T}{v^D - v^T} - 1 \right) > 0
\end{aligned}$$

3. Fall : $t \in T_1(p_{\sigma(k+1)}) (\Rightarrow t \in T_2(p_{\sigma(k)}))$

$$\begin{aligned}
& t_{\sigma(k+1)}^m \left(t_{\sigma(k)}^m(t) \right) - t_{\sigma(k)}^m \left(t_{\sigma(k+1)}^m(t) \right) \\
&= t_{\sigma(k+1)}^m \left(\frac{2p_{\sigma(k)} - 2x_{turn}}{v^D - v^T} + t \right) \\
&\quad - t_{\sigma(k)}^m \left(\frac{2p_{\sigma(k+1)} - 2x_0}{v^D + v^T} + t \left(\frac{v^D - v^T}{v^D + v^T} \right) \right) \\
&= \frac{2p_{\sigma(k+1)} + 2x_0 - 4x_{turn}}{v^D - v^T} + \left(\frac{2p_{\sigma(k)} - 2x_{turn}}{v^D - v^T} + t \right) \left(\frac{v^D + v^T}{v^D - v^T} \right) \\
&\quad - \left(\frac{2p_{\sigma(k)} - 2x_{turn}}{v^D - v^T} + \frac{2p_{\sigma(k+1)} - 2x_0}{v^D + v^T} + t \left(\frac{v^D - v^T}{v^D + v^T} \right) \right) \\
&= p_{\sigma(k+1)} \left(\frac{4v^T}{(v^D - v^T)(v^D + v^T)} \right) + p_{\sigma(k)} \left(\frac{4v^T}{(v^D - v^T)^2} \right) \\
&\quad + x_0 \left(\frac{4v^D}{(v^D - v^T)(v^D + v^T)} \right) + x_{turn} \left(\frac{-2(v^D - v^T) - 2(v^D + v^T)}{(v^D - v^T)^2} \right) \\
&\quad + t \left(\frac{(v^D + v^T)^2 - (v^D - v^T)^2}{(v^D - v^T)(v^D + v^T)} \right) \\
&> p_{\sigma(k+1)} \left(\frac{4v^T}{(v^D - v^T)(v^D + v^T)} \right) + p_{\sigma(k)} \left(\frac{4v^T}{(v^D - v^T)^2} \right) \\
&\quad + x_0 \left(\frac{4v^D}{(v^D - v^T)(v^D + v^T)} \right) + x_{turn} \left(\frac{-4v^D}{(v^D - v^T)^2} \right) \\
&\stackrel{4.15}{>} p_{\sigma(k+1)} \left(\frac{4v^T}{(v^D - v^T)(v^D + v^T)} \right) \\
&\quad + \left(\frac{(x_{turn} - x_0)(v^D - v^T)}{v^T} + x_{turn} \right) \left(\frac{4v^T}{(v^D - v^T)^2} \right)
\end{aligned}$$

$$\begin{aligned}
& + x_0 \left(\frac{4v^D}{(v^D - v^T)(v^D + v^T)} \right) + x_{turn} \left(\frac{-4v^D}{(v^D - v^T)^2} \right) \\
& = p_{\sigma(k+1)} \left(\frac{4v^T}{(v^D - v^T)(v^D + v^T)} \right) + \frac{4x_{turn}v^D}{(v^D - v^T)^2} - \frac{4x_{turn}v^T}{(v^D - v^T)^2} \\
& \quad - \frac{4x_0v^D}{(v^D - v^T)^2} + \frac{4x_0v^T}{(v^D - v^T)^2} + \frac{4x_{turn}v^T}{(v^D - v^T)^2} \\
& \quad + \frac{4x_0v^D}{(v^D - v^T)(v^D + v^T)} - \frac{4v^D x_{turn}}{(v^D - v^T)^2} \\
& = p_{\sigma(k+1)} \left(\frac{4v^T}{(v^D - v^T)(v^D + v^T)} \right) \\
& \quad + \frac{-4x_0v^D(v^D + v^T) + 4x_0v^T(v^D + v^T) + 4x_0v^D(v^D - v^T)}{(v^D - v^T)^2(v^D + v^T)} \\
& = p_{\sigma(k+1)} \left(\frac{4v^T}{(v^D - v^T)(v^D + v^T)} \right) + x_0 \frac{4v^T(v^T - v^D)}{(v^D - v^T)^2(v^D + v^T)} \\
& = (p_{\sigma(k+1)} - x_0) \left(\frac{4v^T}{(v^D - v^T)(v^D + v^T)} \right) > 0
\end{aligned}$$

Für alle drei Fälle ist der Begegnungszeitpunkt früher, nachdem der letzte Kunde gemäß des Schedules σ' bedient wurde. Hierbei ist σ' der Schedule, der durch Vertauschen von $\sigma(k) = n$ und $\sigma(k+1)$ in σ entsteht. Dies impliziert, dass σ kein optimaler Schedule sein kann. \square

Satz 4.4.38. *Das Line-TSPwD ohne Deadlines und ohne Release Dates mit dem Ziel der Minimierung von $\max\{t_i^m, t_i\}$ ist \mathcal{NP} -schwer.*

Beweis. Gegeben sei eine Even-Odd-Partition-Problem-Instanz I mit $x_1, x_2, \dots, x_{2l} \in \mathbb{Z}_{>0}$, so dass $x_i < x_{i+1}$ für alle $i \in \{1, \dots, 2l-1\}$ gilt und $\sum_{i=1}^{2l} x_i = 2q$ ist. Außerdem sei $n = 2l$. Aus diesen Parametern wird nun eine Line-TSPwD-Instanz $\Psi(I)$ mit der Entscheidungsfrage konstruiert: „Existiert eine zulässige Lösung für deren Zielfunktionswert ϑ gilt $\vartheta \leq 10q$?“. Der LKW hat eine Geschwindigkeit von $v^T = 1$, während die Drohne eine Geschwindigkeit von $v^D = 2$ hat. Es gibt $2l + 4$ Kunden mit folgenden Positionen:

$$p_{2l+1} = 12q \quad p_{2l+2} = 9q \quad p_{2l+3} = 0 \quad p_{2l+4} = p_{i^*} = 8q$$

$$p_{2k-a} = \frac{x_{2k-a}}{2} \left(\frac{1}{3} \right)^{l-k} + 9q \quad \forall a \in \{0, 1\} \forall k \in \{1, \dots, l\}.$$

Außerdem wird vorausgesetzt, dass die Kunden $2l+2$ und $2l+3$ nur durch den LKW bedient werden können, also es gilt $T = \{2l+2, 2l+3\}$. Die LKW-Tour kann als fest aufgefasst werden. Aus der Geschwindigkeit des LKWs folgt, dass dieser erst Kunde $2l+2$ und danach Kunde $2l+3$ bedienen kann. Der LKW erreicht Kunde $2l+3$ zum Zeitpunkt $\frac{9q-8q}{v^T} + \frac{9q-0}{v^T} = 10q$.

Es liegt also eine LKW-Tour vor, bei der der LKW strikt in eine Richtung bis zum Wendeort $9q$ fährt, anschließend wendet und strikt in die entgegengesetzte Richtung fährt (siehe Abbildung 4.10). Dies ist also eine LKW-Tour, wie in

4.4.31. Für die Orte der Kunden i mit $i \in \{1, \dots, 2l, 2l+1\}$ gilt $p_i < p_j$ für $i < j$, da

$$\begin{aligned} p_{2k} &= \frac{x_{2k}}{2} \cdot \left(\frac{1}{3}\right)^{l-k} + 9q > \frac{x_{2k-1}}{2} \cdot \left(\frac{1}{3}\right)^{l-k} + 9q = p_{2k-1} \\ p_{2k+1} &= p_{2(k+1)-1} = \frac{x_{2k+1}}{2} \cdot \left(\frac{1}{3}\right)^{l-k-1} + 9q > \frac{x_{2k}}{2} \cdot \left(\frac{1}{3}\right)^{l-k} + 9q = p_{2k} \\ p_{2l+1} &= 12q > \frac{x_{2l}}{2} + 9q = \frac{x_{2l}}{2} \left(\frac{1}{3}\right)^{l-l} + 9q = p_{2l} \end{aligned}$$

gilt. Wegen $x_{2l} < 2q$ folgt die letzte Ungleichung. Der LKW wird diese Kunden nicht anfahren können, da $p_i > 9q = x_{turn}$ für alle $i \in \{1, \dots, 2l, 2l+1\}$ gilt. Somit muss die Drohne diese Kunden bedienen.

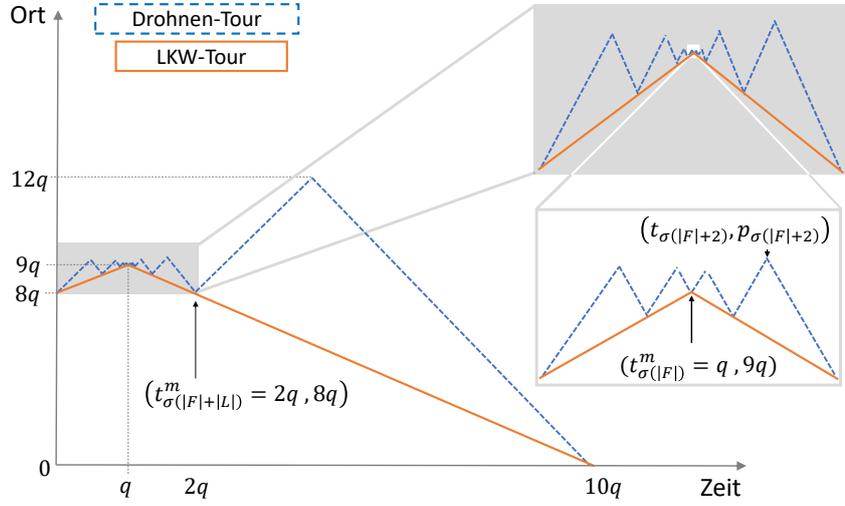
Es gilt

$$p_{2l+1} = 12q > 10q = \frac{(9q - 8q)(2 - 1)}{1} + 9q = \frac{(x_{turn} - x_0)(v^D - v^T)}{v^T} + x_{turn}.$$

Außerdem ist $p_{2l+1} > p_i$ für alle $i \in \{1, \dots, 2l\}$. Folglich wird gemäß 4.4.37 Kunde $2l+1$ in einem optimalen Schedule als letzter Kunde durch die Drohne bedient. Um eine Lösung zu erhalten, die nicht später als $10q$ endet, muss der Trennungszeitpunkt spätestens zum Zeitpunkt $2q \in T_3(p_{2l+1})$ stattfinden:

$$\begin{aligned} t_{2l+1}^m(2q) &= \frac{2p_{2l+1} + 2x_0 - 4x_{turn}}{v^D - v^T} + t \left(\frac{v^T + v^D}{v^D - v^T} \right) \\ &= \frac{24q + 16q - 36q}{2 - 1} + 2q \left(\frac{3}{1} \right) = 10q. \end{aligned}$$

Um zu gewährleisten, dass alle Kunden bedient werden, muss die Drohne die Kunden $1, \dots, 2l$ bis zum Zeitpunkt $2q$ bedient haben. Es sei angenommen, dass σ ein optimaler Drohnen-Schedule bezüglich der Kunden $1, \dots, 2l$ unter der festen LKW-Tour ist. Die Menge der Drohnenkunden lässt sich partitionieren: in eine Menge F von Kunden, deren Begegnungszeitpunkt vor oder zum Zeitpunkt q liegt, und in eine Menge L von Kunden, deren Begegnungszeitpunkte nach q liegen. Dies impliziert, dass $\sigma(|F| + 1)$ der erste Drohnenkunde ist, dessen Begegnungszeitpunkt nach q stattfindet. Entweder ist Kunde $\sigma(|F| + 1)$ ein Wendekunde oder sein Trennungszeitpunkt ist q , falls kein Wendekunde existiert. In Abbildung 4.10 wird das Szenario dargestellt, bei dem es keinen Wendekunden gibt. Im Folgenden wird unter anderem gezeigt, dass eine *Ja*-Antwort impliziert, dass es keinen Wendekunden gibt. Darüber hinaus sind in Abbildung 4.10 die Zeitpunkte $t_{\sigma(|F|)}^m$ und $t_{\sigma(n)}^m$ dargestellt, welche den Begegnungszeitpunkt des letzten Kunden der Menge F und den Begegnungszeitpunkt des letzten Kunden der Menge L kennzeichnen. Der folgende Ausdruck des Zeitpunkts q lässt sich mithilfe von Lemma 4.4.23 unter Berücksichtigung des Drohnen-Schedules $(\sigma(1), \dots, \sigma(|F|))$ bestimmen.

Abbildung 4.10: Lösung der Line-TSPwD-Instanz $\Psi(I)$ **Behauptung 1.**

$$q = \left(q - t_{\sigma(|F|)}^m \right) \cdot \left(\frac{2+1}{2-1} \right)^{|F|} + 2 \left(\sum_{k=1}^{|F|} \frac{p_{\sigma(k)} - x_{turn}}{2-1} \cdot \left(\frac{2+1}{2-1} \right)^{k-1} \right).$$

Dies wird nun gezeigt. Nach 4.4.4 gilt

$$t_{\sigma(|F|)}^m = 2 \left(\sum_{k=0}^{|F|-1} \left(\frac{v^D - v^T}{v^D + v^T} \right)^k \frac{p_{\sigma(|F|-k)} - x_0}{v^D + v^T} \right)$$

Wegen 4.4.23 und $x_{turn} = v^T q + x_0 \Leftrightarrow x_0 = x_{turn} - v^T q$ ist

$$t_{\sigma(|F|)}^m = 2 \left(\sum_{k=0}^{|F|-1} \left(\frac{v^D + v^T}{v^D - v^T} \right)^k \frac{p_{\sigma(k+1)} - v^T t_{\sigma(|F|)}^m + v^T q - x_{turn}}{v^D - v^T} \right).$$

Somit ist mit 4.4.21

$$\begin{aligned} t_{\sigma(|F|)}^m &= 2 \left(\sum_{k=0}^{|F|-1} 3^k \frac{p_{\sigma(k+1)} - x_{turn}}{v^D - v^T} \right) + 2 \left(\sum_{k=0}^{|F|-1} 3^k \frac{v^T q - v^T t_{\sigma(|F|)}^m}{v^D - v^T} \right) \\ \Leftrightarrow t_{\sigma(|F|)}^m - 2 \left(\sum_{k=0}^{|F|-1} 3^k \frac{v^T q - v^T t_{\sigma(|F|)}^m}{v^D - v^T} \right) &= 2 \left(\sum_{k=0}^{|F|-1} 3^k \frac{p_{\sigma(k+1)} - x_{turn}}{v^D - v^T} \right) \\ \Leftrightarrow t_{\sigma(|F|)}^m - 2 \left(\left(\frac{v^T q - v^T t_{\sigma(|F|)}^m}{v^D - v^T} \right) \left(\sum_{k=0}^{|F|-1} 3^k \right) \right) &= 2 \left(\sum_{k=0}^{|F|-1} 3^k \frac{p_{\sigma(k+1)} - x_{turn}}{v^D - v^T} \right) \end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow t_{\sigma(|F|)}^m - 2 \left(\left(\frac{v^T q - v^T t_{\sigma(|F|)}^m}{v^D - v^T} \right) \left(\frac{3^{|F|} - 1}{3 - 1} \right) \right) = 2 \left(\sum_{k=0}^{|F|-1} 3^k \frac{p_{\sigma(k+1)} - x_{turn}}{v^D - v^T} \right) \\
&\Leftrightarrow t_{\sigma(|F|)}^m - (q - t_{\sigma(|F|)}^m) 3^{|F|} + q - t_{\sigma(|F|)}^m = 2 \left(\sum_{k=0}^{|F|-1} 3^k \frac{p_{\sigma(k+1)} - x_{turn}}{v^D - v^T} \right) \\
&\Leftrightarrow q = (q - t_{\sigma(|F|)}^m) 3^{|F|} + 2 \left(\sum_{k=0}^{|F|-1} 3^k \frac{p_{\sigma(k+1)} - x_{turn}}{v^D - v^T} \right).
\end{aligned}$$

Damit ist Behauptung 1 bewiesen.

Im Anschluss erfolgt die Ermittlung des Terms für den Begegnungzeitpunkt des vorletzten Kunden $\sigma(n)$, welcher durch die Drohne bedient wird. Der Beginn des Fluges der Drohne zum Kunden $\sigma(|F| + 1)$ erfolgt zum Zeitpunkt $t_{\sigma(|F|)}^m$. Die Flugdauer, um vom LKW zu diesem Kunden und wieder zurück zum LKW zu fliegen, beträgt $\frac{2p_{\sigma(|F|+1)} - 2x_{turn}}{2-1}$ (siehe Abbildung 4.8). Ab diesem Begegnungszeitpunkt werden dann alle übrigen Kunden der Menge L durch die Drohne bedient, während sich der LKW strikt entfernt. Somit ergibt sich der folgende Begegnungszeitpunkt für $\sigma(n)$:

$$t_{\sigma(n)}^m = t \cdot 3^{|L|-1} + 2 \left(\sum_{k=0}^{|L|-2} 3^k (p_{\sigma(|L|+|F|-k)} - x_{turn}) \right) + q,$$

wobei $t = t_{\sigma(|F|)}^m + \frac{2p_{\sigma(|F|+1)} - 2x_{turn}}{2-1} - q$ ist. Unter Verwendung von Behauptung 1 lässt sich insgesamt der folgende Ausdruck für den Begegnungszeitpunkt von $\sigma(n)$ ableiten:

$$\begin{aligned}
t_{\sigma(n)}^m &= \left(t_{\sigma(|F|)}^m + 2p_{\sigma(|F|+1)} - 2x_{turn} - q \right) \cdot 3^{|L|-1} \\
&\quad + 2 \left(\sum_{k=0}^{|L|-2} 3^k (p_{\sigma(|F|+|L|-k)} - x_{turn}) \right) \\
&\quad + (q - t_{\sigma(|F|)}^m) \cdot 3^{|F|} + 2 \left(\sum_{k=1}^{|F|} (p_{\sigma(k)} - x_{turn}) \cdot 3^{k-1} \right) \\
&= (q - t_{\sigma(|F|)}^m) (3^{|F|} - 3^{|L|-1}) + 2 \left(\sum_{k=0}^{|F|-1} (p_{\sigma(k+1)} - x_{turn}) \cdot 3^k \right) \\
&\quad + 2 \left(\sum_{k=0}^{|L|-1} (p_{\sigma(|F|+|L|-k)} - x_{turn}) \cdot 3^k \right)
\end{aligned} \tag{4.16}$$

Zunächst wird nur der Term aus (4.16) betrachtet ohne den ersten Summanden:
 $\Sigma(\sigma, F, L)$

$$:= 2 \left(\sum_{k=0}^{|F|-1} (p_{\sigma(k+1)} - x_{turn}) \cdot 3^k \right) + 2 \left(\sum_{k=0}^{|L|-1} (p_{\sigma(|F|+|L|-k)} - x_{turn}) \cdot 3^k \right) \quad (4.17)$$

Behauptung 2. $\Sigma(\sigma, F, L)$ ist genau dann minimal, wenn

1. $|F| = |L|$ und
2. $\{\sigma(k+1), \sigma(n-k)\} = \{2(l-k), 2(l-k)-1\} \forall k \in \{0, \dots, l-1\}$

erfüllt sind. Darüber hinaus ist das Minimum gleich $2q$.

Dies kann (Sedding, 2020) entnommen werden. Da dort jedoch einige Aussagen nicht zu finden sind, welche zum Beweis von Behauptung 3 noch benötigt werden, wird Behauptung 2 hier nochmal vollständig bewiesen. Zunächst wird die Implikation gezeigt, dass $|F| = |L|$ gilt, falls $\Sigma(\sigma, F, L)$ minimal ist. O.B.d.A. sei $|L| > |F|$ angenommen. Es wird nun eine neue Lösung konstruiert, bei der nun der erste Kunde aus L bezüglich σ , der Menge F hinzugefügt wird. Der Schedule σ bleibt unverändert. Die beiden neuen Mengen sind demnach

$$\tilde{F} := F \cup \{\sigma(|F|+1)\} \text{ und } \tilde{L} := L \setminus \{\sigma(|F|+1)\}. \quad (4.18)$$

Daraus folgt

$$\begin{aligned} \Sigma(\sigma, \tilde{F}, \tilde{L}) &= 2 \left(\sum_{k=0}^{|\tilde{F}|-1} (p_{\sigma(k+1)} - x_{turn}) \cdot 3^k \right) + 2 \left(\sum_{k=0}^{|\tilde{L}|-1} (p_{\sigma(n-k)} - x_{turn}) \cdot 3^k \right) \\ &= 2 \left(\sum_{k=0}^{|F|-1} (p_{\sigma(k+1)} - x_{turn}) \cdot 3^k + (p_{\sigma(|\tilde{F}|)} - x_{turn}) 3^{|\tilde{F}|} \right) \\ &\quad + 2 \left(\sum_{k=0}^{|L|-1} (p_{\sigma(n-k)} - x_{turn}) \cdot 3^k - (p_{\sigma(n-|L|+1)} - x_{turn}) 3^{|L|-1} \right) \\ &= 2 (p_{\sigma(|F|+1)} - x_{turn}) (3^{|\tilde{F}|} - 3^{|\tilde{L}|-1}) \\ &\quad + 2 \left(\sum_{k=0}^{|F|-1} (p_{\sigma(k+1)} - x_{turn}) \cdot 3^k \right) + 2 \left(\sum_{k=0}^{|L|-1} (p_{\sigma(n-k)} - x_{turn}) \cdot 3^k \right) \\ &= 2 (p_{\sigma(|F|+1)} - x_{turn}) (3^{|\tilde{F}|} - 3^{|\tilde{L}|-1}) + \Sigma(\sigma, F, L). \end{aligned} \quad (4.19)$$

Es sei angemerkt, dass aus der Gleichung $|F| + |L| = 2l$ und der Bedingung $|F| < |L|$ die Ungleichung

$$|F| < |L| - 1 \quad (4.20)$$

folgt. Sei angenommen, dies ist nicht der Fall, d.h. $|F| = |L| - 1$ ist erfüllt. Dann ist $|L| + |F| = 2|L| - 1 = 2l$. Dies ist aber nicht möglich. Somit folgt mit (4.19) und (4.20)

$$\Sigma(\sigma, \tilde{F}, \tilde{L}) - \Sigma(\sigma, F, L) = 2 (p_{\sigma(|F|+1)} - x_{turn}) (3^{|\tilde{F}|} - 3^{|\tilde{L}|-1}) < 0. \quad (4.21)$$

Dies ist ein Widerspruch zur Minimalität von $\Sigma(\sigma, F, L)$. Daher muss $|F| = |L|$ sein. Für das Minimum von (4.17) ergibt sich $|F| = |L| = n/2$. Damit folgt

$$\begin{aligned} & 2 \left(\sum_{k=0}^{|F|-1} (p_{\sigma(k+1)} - x_{turn}) 3^k \right) + 2 \left(\sum_{k=0}^{|L|-1} (p_{\sigma(n-k)} - x_{turn}) 3^k \right) \\ &= 2 \left(\sum_{k=0}^{(n/2)-1} (p_{\sigma(k+1)} - x_{turn}) 3^k \right) + 2 \left(\sum_{k=0}^{(n/2)-1} (p_{\sigma(n-k)} - x_{turn}) 3^k \right) \\ &= 2 \left(\sum_{k=0}^{(n/2)-1} (p_{\sigma(k+1)} + p_{\sigma(n-k)} - 2x_{turn}) 3^k \right) \end{aligned}$$

Die Folge $(3^k)_{k \in \mathbb{N}}$ ist monoton steigend, und es gilt $p_i < p_{i+1}$ für alle $i \in \{1, \dots, n\}$. Somit wird (4.17) durch die Zuordnung der folgenden Form minimiert:

$$\{\sigma(k+1), \sigma(n-k)\} = \{2(l-k), 2(l-k)-1\}.$$

Daher ist das Minimum von (4.17) gleich

$$\begin{aligned} & 2 \left(\sum_{k=0}^{(n/2)-1} (p_{\sigma(k+1)} + p_{\sigma(n-k)} - 2x_{turn}) 3^k \right) \\ &= 2 \left(\sum_{k=0}^{(n/2)-1} (p_{2((n/2)-k)} + p_{2((n/2)-k)-1} - 2x_{turn}) 3^k \right) \\ &= 2 \left(\sum_{k=0}^{(n/2)-1} \left(\frac{1}{2} x_{2((n/2)-k)} \left(\frac{1}{3} \right)^{(n/2)-(n/2)+k} + 9q \right) \right. \\ & \quad \left. + \left(\left(\frac{1}{2} x_{2((n/2)-k)-1} \right) \left(\frac{1}{3} \right)^{(n/2)-(n/2)+k} + 9q - 18q \right) 3^k \right) \quad (4.22) \\ &= 2 \left(\sum_{k=0}^{(n/2)-1} \frac{1}{2} (x_{2((n/2)-k)} + x_{2((n/2)-k)-1}) \left(\frac{1}{3} \right)^k 3^k \right) \\ &= \sum_{k=0}^{(n/2)-1} x_{n-2k} + x_{n-2k-1} = \sum_{k=1}^{(n/2)} x_{2k} + x_{2k-1} = \sum_{k=1}^n x_k = 2q \end{aligned}$$

Insgesamt wurde eine Implikation von Behauptung 2 bewiesen.

Nun wird die andere Implikation gezeigt. Dazu seien F, L und σ gegeben mit $|F| = |L|$ und $\{\sigma(k+1), \sigma(n-k)\} = \{2(l-k), 2(l-k)-1\}$. Dann ist $\Sigma(\sigma, F, L)$ wegen (4.22) gleich $2q$ und somit minimal. Also folgt Behauptung 2.

Jetzt wird der vollständige Ausdruck von $t_{\sigma(n)}^m$ wieder betrachtet, also die Darstellung aus (4.16). Um sicherzustellen, dass sich die Drohne spätestens zum Zeitpunkt $10q$ mit dem LKW trifft und alle Kunden $1, \dots, 2l$ vorab bedient wurden, muss die Ungleichung $2q \geq t_{\sigma(n)}^m$ erfüllt sein. Nun soll die folgende Äquivalenz bewiesen werden.

Behauptung 3.

$$t_{\sigma(n)}^m \leq 2q \Leftrightarrow |F| = |L| \wedge q - t_{\sigma(|F|)}^m = 0 \\ \wedge \{\sigma(k+1), \sigma(n-k)\} = \{2(l-k), 2(l-k)-1\} \\ \forall k \in \{0, \dots, l-1\}$$

Hierbei bedeutet $q - t_{\sigma(|F|)}^m = 0$, dass es keinen Wendekunden gibt. Im Folgenden wird zwischen den drei Fällen $|F| > |L|$, $|L| > |F|$ und $|F| = |L|$ unterschieden. Es wird sich zeigen, dass eine zulässige Lösung nur im Fall $|F| = |L|$ möglich ist. Angenommen $|F| > |L|$. Dann ist

$$t_{\sigma(n)}^m = \left(q - t_{\sigma(|F|)}^m \right) \left(3^{|F|} - 3^{|L|-1} \right) + 2 \left(\sum_{k=0}^{|F|-1} (p_{\sigma(k+1)} - x_{turn}) \cdot 3^k \right) \\ + 2 \left(\sum_{k=0}^{|L|-1} (p_{\sigma(n-k)} - x_{turn}) \cdot 3^k \right) > 2q,$$

da nach (4.21) die Summe (4.17) größer als $2q$ ist und $q - t_{\sigma(|F|)}^m \geq 0$ sowie

$$\left(3^{|F|} - 3^{|L|-1} \right) > 0$$

ist. Zulässigkeit ist in diesem Fall also nicht möglich.

Nun wird der Fall $|L| > |F|$ betrachtet. Da Kunde $\sigma(|F|+1)$ der erste ist, dessen Begegnungszeitpunkt strikt nach q liegt, folgt

$$\left(q - t_{\sigma(|F|)}^m \right) < 2 \left(p_{\sigma(|F|+1)} - x_{turn} \right). \quad (4.23)$$

Andernfalls wäre $\sigma(|F|+1)$ der Menge F zugeordnet worden.

Daher gilt die Abschätzung

$$t_{\sigma(n)}^m = \left(q - t_{\sigma(|F|)}^m \right) \left(3^{|F|} - 3^{|L|-1} \right) + \Sigma(\sigma, F, L) \\ = \left(q - t_{\sigma(|F|)}^m \right) \left(3^{|F|} - 3^{|L|-1} \right) + \Sigma(\sigma, F, L) \\ + 2 \left(p_{\sigma(|F|+1)} - x_{turn} \right) \left(3^{|F|} - 3^{|L|-1} \right) \\ - 2 \left(p_{\sigma(|F|+1)} - x_{turn} \right) \left(3^{|F|} - 3^{|L|-1} \right) \\ \stackrel{(4.19)}{=} \left(q - t_{\sigma(|F|)}^m \right) \left(3^{|F|} - 3^{|L|-1} \right) + \Sigma(\sigma, \tilde{F}, \tilde{L}) \\ - 2 \left(p_{\sigma(|F|+1)} - x_{turn} \right) \left(3^{|F|} - 3^{|L|-1} \right) \\ \stackrel{(4.22)}{\geq} \left(q - t_{\sigma(|F|)}^m \right) \left(3^{|F|} - 3^{|L|-1} \right) + 2q \\ - 2 \left(p_{\sigma(|F|+1)} - x_{turn} \right) \left(3^{|F|} - 3^{|L|-1} \right) \\ \stackrel{(4.23)}{>} 2 \left(p_{\sigma(|F|+1)} - x_{turn} \right) \left(3^{|F|} - 3^{|L|-1} \right) + 2q \\ - 2 \left(p_{\sigma(|F|+1)} - x_{turn} \right) \left(3^{|F|} - 3^{|L|-1} \right) = 2q.$$

Somit liegt für $|F| > |L|$ und $|L| > |F|$ der Begegnungszeitpunkt $t_{\sigma(n)}^m$ des letzten Kunden später als $2q$.

Sei $|F| = |L|$. So lässt sich $t_{\sigma(n)}^m$ mithilfe von (4.22) wie folgt abschätzen:

$$\begin{aligned} t_{\sigma(n)}^m &= \left(q - t_{\sigma(|F|)}^m \right) \left(3^{|F|} - 3^{|L|-1} \right) + \Sigma(\sigma, F, L) \\ &\geq \left(q - t_{\sigma(|F|)}^m \right) \left(3^{(n/2)} \left(1 - \left(\frac{1}{3} \right) \right) \right) + 2q. \end{aligned}$$

Da $(3^{(n/2)}(1 - (\frac{1}{3}))) > 0$ ist und vorausgesetzt wurde, dass $q - t_{\sigma(|F|)}^m \geq 0$ ist, liegt genau dann Zulässigkeit vor, wenn $q - t_{\sigma(|F|)}^m = 0$ gilt. Damit ist die Implikation " \Rightarrow " bewiesen. Die Implikation " \Leftarrow " von Behauptung 3 gilt ebenfalls, da gezeigt wurde, dass durch das Einsetzen der Voraussetzungen ein Begegnungszeitpunkt von $2q$ resultiert. Also gilt Behauptung 3.

Im letzten Schritt wird gezeigt, dass I genau dann mit *Ja* beantwortet wird, wenn $\Psi(I)$ mit *Ja* beantwortet wird. Es wurde gezeigt, dass die Instanz $\Psi(I)$ genau dann mit *Ja* beantwortet wird, wenn die rechte Seite von Behauptung 3 erfüllt ist, d.h.

$$\begin{aligned} &|F| = |L| \\ &\text{und } q - t_{\sigma(|F|)}^m = 0 \\ &\text{und } \{\sigma(k+1), \sigma(n-k)\} = \{2(l-k), 2(l-k)-1\} \quad \forall k \in \{0, \dots, l-1\}. \end{aligned} \quad (4.24)$$

Damit und mit Behauptung 1 folgt

$$\begin{aligned} q &= \left(q - t_{\sigma(|F|)}^m \right) \cdot 3^{|F|} + 2 \left(\sum_{k=1}^{|F|} (p_{\sigma(k)} - x_{turn}) \cdot 3^{k-1} \right) \\ &= 2 \left(\sum_{k=1}^{n/2} (p_{\sigma(k)} - x_{turn}) \cdot 3^{k-1} \right) = 2 \left(\sum_{k=1}^{n/2} (p_{\sigma((n/2)-(k-1))} - x_{turn}) \cdot 3^{(n/2)-k} \right) \\ &= \sum_{k=1}^{n/2} x_{\sigma((n/2)-(k-1))} = \sum_{k=1}^{n/2} x_{\sigma(k)}. \end{aligned}$$

Somit ist (4.24) äquivalent zu

$$\begin{aligned} &|F| = |L| \\ &\text{und } q = \sum_{k=1}^{n/2} x_{\sigma(k)} \\ &\text{und } \{\sigma(k+1), \sigma(n-k)\} = \{2(l-k), 2(l-k)-1\} \quad \forall k \in \{0, \dots, l-1\}, \end{aligned}$$

das einer zulässigen Lösung von I entspricht. \square

Satz 4.4.39. *Das Line-TSPwD ohne Deadlines und Release Dates mit dem Ziel der Minimierung von $\max\{t_i\}$ ist \mathcal{NP} -schwer.*

Beweis. Der Beweis ist analog zum Beweis von 4.4.38. Nur die Anpassung von $p_{2l+1} = 24q$ ist erforderlich. Daher ist weiterhin der Kunde $2l+1$ der entfernteste Kunde, der durch die Drohne bedient werden muss. Da

$$0 \in T_2(p_{2l+1}) = \left(\left(\frac{2+1}{2-1} \right) \left(\frac{9q-8q}{1} - \left(\frac{48q}{1} - \frac{16q}{3} \right) \right), \frac{9q-8q}{1} \right] = (-125q, q]$$

gilt, ist

$$t_{2l+1}^m(0) = \frac{2p_{2l+1} - 2x_{turn}}{v^T - v^D} = \frac{48q - 18q}{2 - 1} = 30q > 10q.$$

Der Kunde $2l + 1$ muss als letzter durch die Drohne bedient werden, damit alle Kunden zum Zeitpunkt $10q$ bedient werden können. Startet die Drohne zum Zeitpunkt $2q$, so ergibt sich für die Drohnen-Tour die Funktion $f^D(t) = 2t + 4q$. Der Kunde $2l + 1$ wird zum Zeitpunkt $10q$ bedient, wenn die Drohne zum Zeitpunkt $2q$ vom LKW startet, da

$$f^D(10q) = 2 \cdot 10q + 4q = 24q.$$

Der verbleibende Beweis ist identisch mit dem Beweis in 4.4.38. \square

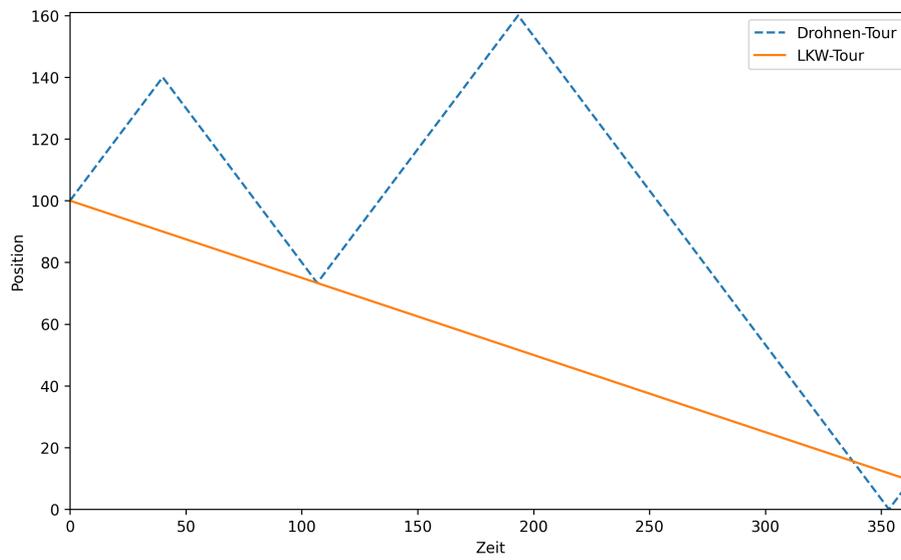
Korollar 4.4.40. *Sowohl das Line-TSPwD mit schnellerer Drohne ohne Zeitfenster, bei dem eine Rückkehr zum Depot erforderlich ist, als auch das MDRP in (Poikonen & Golden, 2020) sind \mathcal{NP} -schwer.*

Beweis. Der Beweis ist identisch zum Beweis in Satz 4.4.38, jedoch werden die Kunden p_{l+1}, p_{l+2} und p_{l+3} nicht benötigt. Der Start ist bei $orig = p_{i^*}$ und das Ziel ist $dest = p_{i^*}$. Die Reichweite der Drohne wird hier auf ∞ gesetzt. \square

Da es sich hier um eine Reduktion ausgehend von einem im schwachen Sinne \mathcal{NP} -schweren Problem, nämlich dem Even-Odd-Partition-Problem, handelt, kann daraus nicht geschlossen werden, dass das Problem \mathcal{NP} -schwer im strengen Sinne ist. Da auch kein pseudopolynomieller Ansatz konstruiert wurde, kann auch nicht auf eine schwache \mathcal{NP} -Schwere geschlossen werden. Diese Frage bleibt in der vorliegenden Arbeit offen.

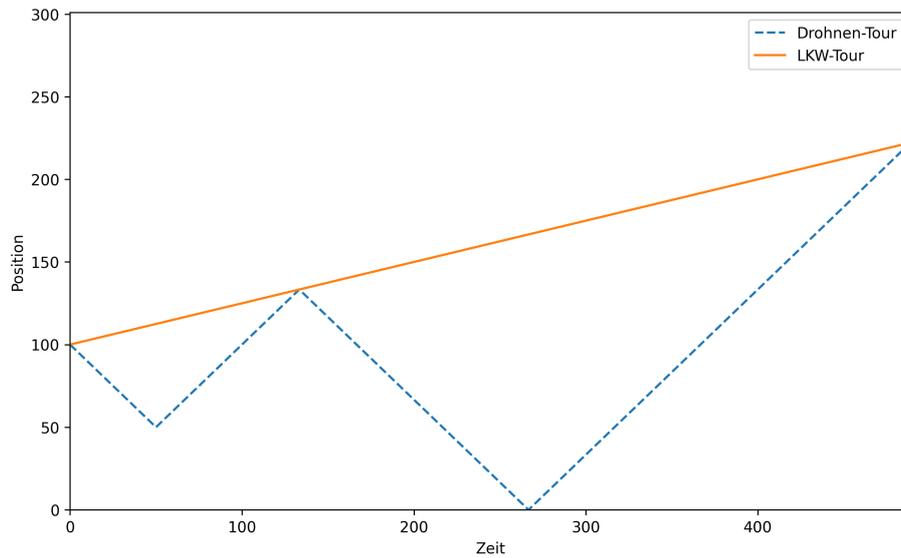
Falls $T = \emptyset$ erfüllt ist, kann man vermuten, dass eine optimale Lösung existiert, bei der der LKW nur in eine Richtung fährt, wie in Beispiel 4.4.24. Jedoch ist dies nicht der Fall, wie das folgende Gegenbeispiel zeigt:

Sei eine Instanz mit 5 Kunden gegeben, die alle durch die Drohne bedient werden können. Die LKW-Geschwindigkeit ist $v^T = 1/4$ und die Drohnen-Geschwindigkeit ist $v^D = 1$. Der Startkunde ist $i^* = 3$. Fährt der LKW strikt nach links, so resultiert die folgende optimale Lösung:



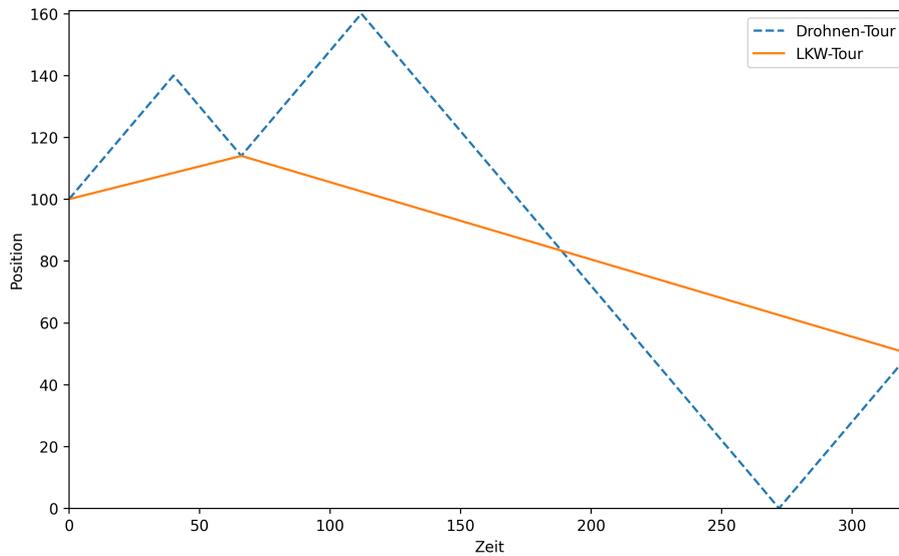
Kunde i	1	2	3	4	5
Position	0	50	100	140	160
t_i	353.3	200.0	0.0	40.0	193.3
r_i^m	9.3	-	-	73.3	15.6
t_i^m	362.7	-	-	106.7	337.8
r_i^s	15.6	-	-	100.0	73.3
t_i^s	337.8	-	-	0.0	106.7

Fährt der LKW strikt nach rechts, so ergibt die folgende optimale Lösung:



Kunde i	1	2	3	4	5
Position	0	50	100	140	160
t_i	266.7	50.0	0.0	160	240
r_i^m	222.2	133.3	-	-	-
t_i^m	488.9	133.3	-	-	-
r_i^s	133.3	100.0	-	-	-
t_i^s	133.3	0.0	-	-	-

Ohne die Beschränkung, dass der LKW nur in eine Richtung fährt, resultiert die folgende optimale Lösung:



Kunde i	1	2	3	4	5
Position	0	50	100	140	160
t_i	272	322	0	40	112
r_i^m	50.0	50.0	-	114.0	83.3
t_i^m	322.0	322.0	-	66.0	188.7
r_i^s	83.3	50.0	-	100.0	114.0
t_i^s	188.7	322.0	-	0.0	66.0

Hier wurde nur gezeigt, dass das Problem \mathcal{NP} -schwer ist. Eine schwache oder strenge \mathcal{NP} -Schwere wurde nicht ausdifferenziert. Wenn jedoch Deadlines und Release Dates ins Kalkül miteinbezogen werden, ist dies jedoch möglich. Diese Fälle werden in den folgenden Abschnitten behandelt.

4.5 Komplexität mit Deadlines und $v^D > v^T$

Das Scheduling-Problem $1|p_i = a_i - wt, d_i|C_{\max}$ mit $1 > w > 0$ und $wd_i < a_i \leq d_i$ ist \mathcal{NP} -schwer im strengen Sinne (Cheng & Ding, 1999)(Theorem 1). Die Prozesszeiten hängen von der Zeit t ab. Die Parameter $a_i > 0$ sind die individuellen Basisprozesszeiten. Der Auftrag i muss vor oder zum Zeitpunkt d_i bearbeitet worden sein. In einer Line-TSPwD-Instanz mit LKW-Tour $f^T(t) = v^T \cdot t + x_0$ wurde in Abschnitt 4.4 ermittelt, dass die Flugdauer der Drohne, um

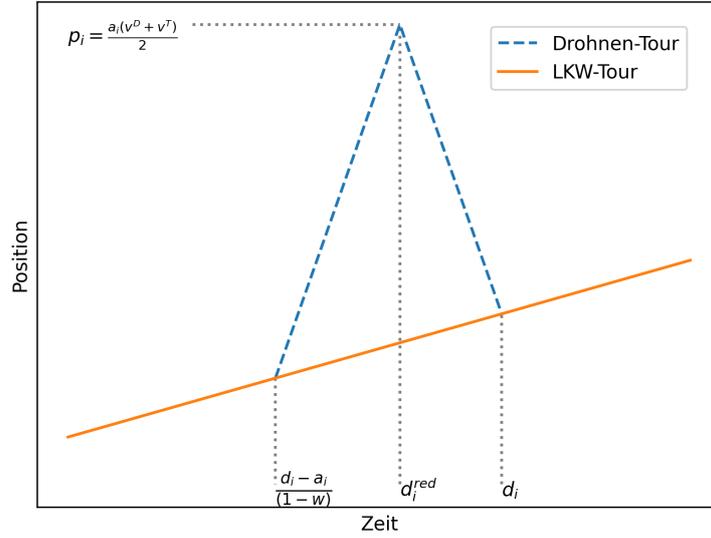


Abbildung 4.11: Spätester Trennungszeitpunkt in $\Psi(I)$, um Kunde i zu seiner Deadline zu bedienen

vom LKW zu starten, den Kunden am Ort $p_k > x_0$ zu bedienen und zurück zum LKW zu fliegen, gleich

$$\frac{2p_k - 2x_0}{v^T + v^D} + t \left(\frac{v^D - v^T}{v^T + v^D} - 1 \right)$$

ist. Somit kann die Flugzeit der Drohne für eine LKW-Tour f^T als eine Prozesszeit gemäß $1|p_i = a_i - wt, d_i|C_{\max}$ aufgefasst werden.

Satz 4.5.1. *Das Line-TSPwD mit beliebigen Deadlines und dem Ziel der Minimierung von $\max\{t_i^m, t_i\}$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Sei eine Instanz I des Problems $1|p_i = a_i - wt, d_i|C_{\max}$ mit $0 < w < 1$ und $wd_i < a_i \leq d_i$ mit n Aufträgen gegeben sowie die Entscheidungsfrage „Gibt es eine zulässige Lösung, deren Zielfunktionswert den Wert Φ nicht übersteigt?“. Nun werden die Parameter für eine Instanz $\Psi(I)$ des Line-TSPwD konstruiert:

$$\begin{aligned} v^T &= 1, & v^D &= \frac{2-w}{w}, & p_{i^*} &= 0, \\ p_i &= \frac{a_i(v^D + v^T)}{2}, & d_i^{red} &= d_i \cdot \left(1 + \frac{v^T}{v^D} \right) - \frac{p_i}{v^D}. \end{aligned}$$

Da $v^T d_i^{red} = d_i^{red} < p_i$ für alle $i \in \{1, \dots, n\}$ gilt, kann der LKW keinen der Kunden rechtzeitig bedienen. Nach Voraussetzung ist $a_i > d_i w$ und somit gilt:

$$a_i > d_i w = d_i \left(\frac{2w}{2} \right) = d_i \left(\frac{2}{\left(\frac{2-w}{w} + 1 \right)} \right) = d_i \left(\frac{2}{\left(\frac{2-w}{w} + 1 \right)} \right) = d_i \left(\frac{2}{v^D + 1} \right)$$

$$\begin{aligned}
&= d_i \left(\frac{2(v^D + v^T)}{(v^D + v^T)(v^D + 1)} \right) = d_i \left(\frac{\left(1 + \frac{v^T}{v^D}\right) 2v^D}{v^D(v^D + v^T) + v^D + v^T} \right) \\
&= d_i \left(\frac{\left(1 + \frac{v^T}{v^D}\right)}{\frac{v^D(v^D + v^T) + v^D + v^T}{2v^D}} \right) \\
&\Leftrightarrow a_i \left(\frac{v^D(v^D + v^T) + v^D + v^T}{2v^D} \right) > d_i \left(1 + \frac{v^T}{v^D} \right) \\
&\Leftrightarrow \frac{a_i(v^D + v^T)}{2} > d_i \left(1 + \frac{v^T}{v^D} \right) - \frac{a_i(v^D + v^T)}{2v^D} \\
&\Leftrightarrow p_i > d_i \left(1 + \frac{v^T}{v^D} \right) - \frac{p_i}{v^D} = d_i^{red} \Leftrightarrow p_i > v^T d_i^{red} = f^T(d_i^{red})
\end{aligned}$$

Die Drohne muss alle Kunden bedienen, damit eine zulässige Lösung zustande kommt.

Die Prozesszeit eines Auftrags der Instanz in $1|a_i - wt, d_i|C_{\max}$ entspricht der Zeit, welche die Drohne benötigt, um zum Zeitpunkt t vom LKW zum entsprechenden Kunden in $\Psi(I)$ zu fliegen und anschließend wieder zurück zum LKW:

$$\begin{aligned}
t_i^m(t) - t &= \frac{2p_i}{v^D + v^T} - t \left(1 - \left(\frac{v^D - v^T}{v^D + v^T} \right) \right) = \frac{2 \left(\frac{a_i(v^D + v^T)}{2} \right)}{v^D + v^T} \\
&\quad - t \left(1 - \frac{\frac{2-w}{w} - 1}{\frac{2-w}{w} + 1} \right) = a_i - t(1 - (1 - w)) = a_i - tw
\end{aligned}$$

Die Drohne muss spätestens zum Zeitpunkt $t = \frac{d_i - a_i}{(1-w)}$ starten, um Kunde i rechtzeitig bedienen zu können. Dies ist in Abbildung 4.11 dargestellt.

$$\begin{aligned}
0 &= d_i \left(\frac{\left(\frac{-2v^D - 2 + 2v^D + 2}{v^D + 1} \right)}{(1-w)v^D} \right) + \frac{a_i(v^D + 1)}{v^D} - \frac{a_i(v^D + 1)(v^D - 1)}{(v^D - 1)v^D} \\
&= d_i \left(\frac{-2 + v^D \left(\frac{2}{v^D + 1} \right) + \left(\frac{2}{v^D + 1} \right)}{(1-w)v^D} \right) + \frac{a_i(v^D + 1)}{v^D} + \frac{a_i(1 - v^D)}{\left(\frac{v^D - 1}{v^D + 1} \right) v^D} \\
&= d_i \left(\frac{-2 + v^D w + w}{(1-w)v^D} \right) + \frac{2(a_i(v^D + 1))}{2v^D} + a_i \left(\frac{1 - v^D}{\left(1 - \left(\frac{2}{v^D + 1} \right) \right) v^D} \right) \\
&= d_i \left(\frac{-1 + v^D - v^D(1-w) - (1-w)}{(1-w)v^D} \right) + \frac{2p_i}{v^D} + a_i \left(\frac{1}{(1-w)v^D} - \frac{1}{(1-w)} \right) \\
&= d_i \left(-\frac{1}{(1-w)v^D} + \frac{1}{(1-w)} - 1 - \frac{1}{v^D} \right) + \frac{2p_i}{v^D} + a_i \left(\frac{1}{(1-w)v^D} - \frac{1}{(1-w)} \right)
\end{aligned}$$

$$\begin{aligned} &\Leftrightarrow \frac{\left(p_i - \frac{d_i - a_i}{(1-w)}\right)}{v^D} + \frac{d_i - a_i}{(1-w)} = d_i \left(1 + \frac{1}{v^D}\right) - \frac{p_i}{v^D} \\ &\Leftrightarrow \frac{\left(p_i - f^T \left(\frac{d_i - a_i}{(1-w)}\right)\right)}{v^D} + \frac{d_i - a_i}{(1-w)} = d_i^{red}. \end{aligned}$$

Analog gilt für die Fertigstellung eines Auftrags der Instanz I das Folgende:

$$\begin{aligned} \frac{d_i - a_i}{(1-w)} + a_i - w \left(\frac{d_i - a_i}{(1-w)}\right) &= \frac{d_i - a_i + a_i(1-w) - wd_i + wa_i}{(1-w)} \\ &= \frac{d_i - wd_i}{(1-w)} = d_i. \end{aligned}$$

Die spätesten Zeitpunkte, zu denen ein Auftrag bzw. die Drohne gestartet werden kann, um Kunden rechtzeitig zu bedienen, sind identisch. Diese ergeben sich aus $\frac{d_i - a_i}{(1-w)}$, wie oben gezeigt wurde. Somit existiert eine zulässige Lösung mit Zielfunktionswert Φ für eine Instanz in $1|p_i = a_i + wt, d_i|C_{\max}$ genau dann, wenn eine zulässige Lösung mit Zielfunktionswert Φ für das Line-TSPwD existiert. Die Transformation erfüllt die Eigenschaften von 4.1.14 und ist somit pseudo-polynomiell. \square

Selbst die Konstruktion einer zulässigen Lösung ist mit der Existenz von Deadlines schwierig:

Satz 4.5.2. *Die Konstruktion einer zulässigen Lösung für Line-TSPwD mit Deadlines ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Sei eine Instanz I für $1|p_i = a_i - wt, d_i|C_{\max}$ mit der Entscheidungsfrage „Lässt sich eine zulässige Lösung finden, deren Zielfunktionswert nicht größer als C ist?“ gegeben. Diese wird nun in eine Line-TSPwD Instanz $\Psi(I)$ mit Deadlines überführt. Die Entscheidungsfrage lautet „Existiert eine zulässige Lösung?“. Im Weiteren wird auf die in 4.5.1 konstruierte Reduktion zurückgegriffen und ein weiterer Kunde $n+1$ mit der folgenden Position und Deadline hinzugefügt:

$$p_{n+1} = C + d_{\max}^{red} \cdot \frac{1}{w}, \quad d_{n+1}^{red} = \frac{d_{\max}^{red}}{2-w},$$

wobei $d_{\max}^{red} = \max\{d_i^{red} | i \in \{1, \dots, n\}\}$ gilt. Für eine Ja-Antwort bezüglich $\Psi(I)$, muss Kunde $n+1$ zuletzt bedient werden, da ansonsten die Deadlines aller anderen Kunden überschritten würden:

$$\begin{aligned} t_{n+1}^m(0) &= \frac{2p_{n+1}}{v^T + v^D} = \frac{2C + 2d_{\max}^{red}}{(v^T + v^D)w} \\ &= \frac{2C}{2w} + \frac{2d_{\max}^{red}}{2w} = C + d_{\max}^{red} > d_k^{red} \quad \forall k \in \{1, \dots, n\} \end{aligned}$$

Die Drohne muss spätestens zum Zeitpunkt C gestartet werden, um Kunde $n+1$ rechtzeitig bedienen zu können:

$$t_{n+1}(C) = \frac{p_{n+1} - C}{v^D} + C = \frac{C + d_{\max}^{red} \frac{1}{w} - C}{v^D}$$

$$= \frac{d_{\max}^{\text{red}}}{wv^D} = \frac{d_{\max}^{\text{red}}}{2-w} = d_{n+1}^{\text{red}}.$$

Somit müssen alle übrigen Kunden bis zum Zeitpunkt C zulässig bedient werden, damit eine zulässige Lösung existiert. Die Transformation erfüllt die Eigenschaften von 4.1.14 und ist somit pseudo-polynomiell. \square

Satz 4.5.3. *Das Line-TSPwD mit beliebigen Deadlines und dem Ziel der Minimierung von $\max t_i$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Auch hier kann wieder auf den Beweis von 4.5.1 verwiesen werden. Zusätzlich wird auch die Instanz um den Kunden $n + 1$ aus dem Beweis von 4.5.2 erweitert. Auch dieser Kunde muss zuletzt bedient werden, wobei sich der Schwellwert zu d_{n+1}^{red} aus 4.5.2 ändert. \square

Korollar 4.5.4. *Das Line-TSPwD mit Deadlines und Release Dates ist sowohl für die Minimierung von $\max\{t_i^m, t_i\}$ als auch für die Minimierung von $\max\{t_i\}$ \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Dies folgt unmittelbar aus 4.5.1. \square

4.6 Komplexität mit Release Dates und $v^D > v^T$

In (Cheng & Ding, 1998) (Theorem 5) wurde die strenge \mathcal{NP} -Schwere des Schedulingproblems $1|p_i = a_i + wt, r_i|C_{\max}$ gezeigt. In diesem Problem liegen also konstante individuelle Prozesszeiten $a_i > 0$ für jeden Kunden vor. Zusätzlich sind Prozesszeiten wt zu berücksichtigen, die von der Startzeit der Bearbeitung t abhängen. Die Steigung $w > 0$ ist für jeden Auftrag konstant. Der Auftrag i kann erst ab Zeitpunkt r_i bearbeitet werden.

Im vorigen Abschnitt wurden Deadlines vorausgesetzt. Dort wurde für die Reduktion das Schedulingproblem $1|p_i = a_i - wt, d_i|C_{\max}$ verwendet. Befinden sich alle Kunden auf einer Seite des Startortes, fährt der LKW strikt in Richtung der Kunden, die von der Drohne bedient werden. Damit entspricht die Flugzeit der Drohne genau den Prozesszeiten des Schedulingproblems. Daher ist das Problem \mathcal{NP} -schwer im strengen Sinne, selbst wenn der Startkunde ganz links oder ganz rechts liegt.

Da nun aber für $1|p_i = a_i + wt, r_i|C_{\max}$ die Prozesszeiten linear ansteigen, muss für das Line-TSPwD eine Instanz konstruiert werden, die eine LKW-Tour erzwingt, sodass sich der LKW von den Kunden entfernt, welche die Drohne bedient. Bei der LKW-Tour $f^T(t) = -v^T \cdot t + x_0$ ergibt sich eine Flugdauer von

$$\frac{2p_k - 2x_0}{v^D - v^T} + t \left(\frac{v^D + v^T}{v^D - v^T} - 1 \right).$$

In diesem Zusammenhang sei nochmals auf Abbildung 4.8 verwiesen.

Das Problem $1|p_i = a_i - wt, r_i|C_{\max}$ ist dagegen polynomiell lösbar, wie in (Cheng & Ding, 1998) gezeigt wird. Dort wird ein Algorithmus präsentiert, dessen Laufzeit in $\mathcal{O}(n^6 \log n)$ liegt. Daher kann dieses Problem nicht verwendet werden, um eine Reduktion analog zum vorherigen Abschnitt zu konstruieren. Es kann dennoch gezeigt werden, dass für $i^* \in \{1, n\}$ das Line-TSPwD mit Release Dates \mathcal{NP} -schwer im strengen Sinne ist. Für den Beweis wird das folgende Lemma genutzt.

Lemma 4.6.1. *Sei eine Line-TSPwD Instanz mit n Kunden gegeben, wobei $v^D > v^T$ gilt und Release Dates existieren. Es liegt eine feste und zulässige LKW-Tour f^T vor, wobei der LKW keinen der Kunden bedient. Dann ist $(\sigma(1) = i^*, \dots, \sigma(n))$ mit gegebenen $t_{\sigma(k)}^m$ genau dann ein zulässiger Drohnen-Schedule, wenn folgende Bedingungen gelten*

$$t_{i^*}^m = 0 \quad (4.25)$$

$$t_{\sigma(k)}^m \geq \max \left\{ t_{\sigma(k-1)}^m + \frac{|f^T(t_{\sigma(k-1)}^m) - p_{\sigma(k)}|}{v^D}, r_{\sigma(k)} \right\} \\ + \frac{|f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}|}{v^D} \quad \forall k \in \{2, \dots, n\} \quad (4.26)$$

Beweis. Für ein Drohnen-Schedule und der Voraussetzung, dass der LKW keinen Kunden bedient, lässt sich ableiten, dass die Eigenschaften eines partiell zulässigen Vektors gemäß der Definition 3.2.1 erfüllt sind. Daher kann das lineare Programm für einen gegebenen partiell zulässigen Vektor aus Abschnitt 3.2.3 benutzt werden.

Sei also zunächst ein Drohnen-Schedule gegeben, welcher (4.25) und (4.26) erfüllt. Im Rahmen der Festlegungen

$$t_{\sigma(k)} = \max \left\{ t_{\sigma(k-1)}^m + \frac{|f^T(t_{\sigma(k-1)}^m) - p_{\sigma(k)}|}{v^D}, r_{\sigma(k)} \right\} \\ t_{\sigma(k)}^s = t_{\sigma(k-1)}^m \\ r_{\sigma(k)}^m = f^T(t_{\sigma(k)}^m) \\ r_{\sigma(k)}^s = f^T(t_{\sigma(k)}^s)$$

müssen nur noch die Restriktionen (3.36),(3.39),(3.43),(3.44),(3.47) und (3.48) überprüft werden. Die Bedingung (4.25) ist äquivalent zu (3.39). Offensichtlich ist $t_{\sigma(k)} \geq r_{\sigma(k)}$. Dies entspricht (3.36).

Die Ungleichung $t_{\sigma(k)} \geq \frac{|p_{\sigma(k)} - f^T(t_{\sigma(k)}^s)|}{v^D} + t_{\sigma(k)}^s$ entspricht der Nebenbedingung (3.43). Ebenso gelten

$$t_{\sigma(k)}^m \geq \frac{|p_{\sigma(k)} - f^T(t_{\sigma(k)}^m)|}{v^D} + t_{\sigma(k)} \Leftrightarrow (3.44)$$

$$t_{\sigma(k)}^s = t_{\sigma(k-1)}^m = \frac{|f^T(t_{\sigma(k-1)}^m) - f^T(t_{\sigma(k)}^s)|}{v^T} + t_{\sigma(k-1)}^m \Leftrightarrow (3.47)$$

$$t_{\sigma(k)}^m \geq \frac{|f^T(t_{\sigma(k)}^m) - f^T(t_{\sigma(k)}^s)|}{v^T} + t_{\sigma(k)}^s \Leftrightarrow (3.48)$$

Damit ist eine Implikation gezeigt.

Es sei nun eine zulässige Lösung für das Modell unter einer gegebenen Reihen-

folgefestlegung der LKW-Operationen sowie einer festen LKW-Tour gegeben. Angenommen (4.26) ist nicht erfüllt. Es sei zunächst angenommen, dass gilt:

$$t_{\sigma(k)}^m < r_{\sigma(k)} + \frac{|f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}|}{v^D}. \quad (4.27)$$

Im Hinblick auf die Zulässigkeit bezüglich des Modells gelten

$$\begin{aligned} t_{\sigma(k)} &\geq r_{\sigma(k)} \\ \text{und} \quad t_{\sigma(k)}^m &\geq t_{\sigma(k)} + \frac{|f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}|}{v^D}. \end{aligned}$$

Somit gilt insgesamt

$$t_{\sigma(k)}^m \geq r_{\sigma(k)} + \frac{|f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}|}{v^D}.$$

Dies widerspricht der Annahme (4.27). Nun sei angenommen, dass gilt:

$$t_{\sigma(k)}^m < t_{\sigma(k-1)}^m + \frac{|f^T(t_{\sigma(k-1)}^m) - p_{\sigma(k)}|}{v^D} + \frac{|f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}|}{v^D}. \quad (4.28)$$

Zunächst wird gezeigt, dass der folgende Zusammenhang erfüllt ist:

$$\frac{|f^T(t_{\sigma(k)}^s) - p_{\sigma(k)}|}{v^D} + t_{\sigma(k)}^s \geq \frac{|f^T(t_{\sigma(k-1)}^m) - p_{\sigma(k)}|}{v^D} + t_{\sigma(k-1)}^m. \quad (4.29)$$

Wegen $|x| - |y| \leq |x + y|$ gilt

$$\frac{|f^T(t_{\sigma(k-1)}^m) - p_{\sigma(k)}| - |p_{\sigma(k)} - f^T(t_{\sigma(k)}^s)|}{v^D} \leq \frac{|f^T(t_{\sigma(k-1)}^m) - f^T(t_{\sigma(k)}^s)|}{v^D}.$$

Für eine zulässige LKW-Tour gilt

$$t - t' \geq \frac{f^T(t) - f^T(t')}{v^T} \quad \forall t, t' \geq 0 \text{ mit } t > t'.$$

Da die Zulässigkeit der LKW-Tour vorausgesetzt wird und die Drohne schneller als der LKW ist, gilt

$$\frac{|f^T(t_{\sigma(k-1)}^m) - f^T(t_{\sigma(k)}^s)|}{v^D} < \frac{|f^T(t_{\sigma(k-1)}^m) - f^T(t_{\sigma(k)}^s)|}{v^T} \leq t_{\sigma(k)}^s - t_{\sigma(k-1)}^m.$$

Es folgt (4.29). Zusätzlich gilt

$$\begin{aligned} t_{\sigma(k)}^m &\geq \frac{|f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}|}{v^D} + t_{\sigma(k)} \\ &\geq \frac{|f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}|}{v^D} + \frac{|f^T(t_{\sigma(k)}^s) - p_{\sigma(k)}|}{v^D} + t_{\sigma(k)}^s \end{aligned}$$

$$\stackrel{(4.29)}{\geq} \frac{|f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}|}{v^D} + \frac{|f^T(t_{\sigma(k-1)}^m) - p_{\sigma(k)}|}{v^D} + t_{\sigma(k-1)}^m.$$

Dies widerspricht der Annahme (4.28). Somit muss eine zulässige Lösung des Modells (4.25) und (4.26) erfüllen. \square

Satz 4.6.2. *Das Line-TSPwD mit Release Dates und $v^D > v^T$ und $i^* \in \{1, n\}$ mit dem Ziel der Minimierung von $\max\{t_i, t_i^m\}$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Sei eine Instanz I des Problems $1|p_i = a_i + wt, r_i|C_{\max}$ mit n Aufträgen und der Entscheidungsfrage „Lässt sich eine zulässige Lösung finden, deren Zielfunktionswert den Wert C nicht überschreitet?“ gegeben. Sei $a_{\max} = \max_{i \in \{1, \dots, n\}} \{a_i\}$. Dann ist $\Psi(I)$ eine Line-TSPwD Instanz mit den folgenden Parametern:

$$\begin{aligned} v^T &= 1, & v^D &= \frac{2+w}{w}, \\ p_i &= \frac{2a_{\max} - a_i}{w}, & p_{n+1} &= C + \frac{2a_{\max}}{w}, & p_{i^*} &= 0, \\ r_{n+1}^{red} &= C + \frac{2a_{\max}}{w}, & r_i^{red} &= r_i \left(1 + \frac{1}{v^D}\right) + \frac{2a_{\max} - p_i}{v^D} + 2\frac{a_{\max}}{w} \end{aligned}$$

mit $i \in \{1, \dots, n\}$. Die Entscheidungsfrage bezüglich $\Psi(I)$ lautet „Gibt es eine zulässige Lösung, deren Zielfunktionswert $C + \frac{2a_{\max}}{w}$ nicht übersteigt?“. Damit die Frage mit *Ja* beantwortet werden kann, muss der LKW zum Zeitpunkt $r_{n+1}^{red} = C + \frac{2a_{\max}}{w}$ bei $p_{n+1} = C + \frac{2a_{\max}}{w} = \frac{C + \frac{2a_{\max}}{w} - 0}{v^T}$ sein, da dies dem Schwellwert entspricht und erst ab diesem Zeitpunkt Kunde $n+1$ bedient werden kann. In dem Fall, dass die Drohne den Kunden $n+1$ bedient, muss diese anschließend zum LKW zurückfliegen. Ist der LKW zu diesem Zeitpunkt nicht am Ort p_{n+1} , ist eine Flugdauer größer als 0 notwendig, sodass die Lösung keine *Ja*-Antwort implizieren würde. Somit ist die LKW-Tour eine fest vorgegebene Route, auf der sich der LKW ohne Unterbrechung und kontinuierlich von Position 0 zu Position p_{n+1} bewegt. Der LKW befindet sich also zum Zeitpunkt t am Ort $f^T(t) = t$. Da $a_{\max} \geq a_i$ für alle $i \in \{1, \dots, n\}$ ist, gilt $p_i = \frac{2a_{\max} - a_i}{w} \geq 0 = p_{i^*}$. Somit ist $\Psi(I)$ eine Instanz mit der Eigenschaft $i^* \in \{1, n\}$.

Ab dem Zeitpunkt $\frac{2a_{\max}}{w}$ besucht der LKW keinen der Orte p_i mit $i \in \{1, \dots, n\}$. Die Drohne muss die entsprechenden Kunden also bedienen. Der früheste Zeitpunkt, zu dem die Drohne vom LKW aus starten kann, um den Kunden i zu bedienen, sodass keine Wartezeiten für die Drohne entstehen, ist der Zeitpunkt $r_i + \frac{2a_{\max}}{w}$, denn es gilt:

$$\begin{aligned} t_i \left(r_i + \frac{2a_{\max}}{w} \right) &= r_i + \frac{2a_{\max}}{w} + \frac{v^T \left(r_i + \frac{2a_{\max}}{w} \right) - p_i}{v^D} \\ &= \frac{2a_{\max}}{w} + \frac{2a_{\max} - p_i}{v^D} + r_i \left(1 + \frac{1}{v^D} \right) = r_i^{red}. \end{aligned}$$

Folglich genügt es, sich auf Lösungen zu beschränken, bei denen die Drohne frühestens zum Zeitpunkt $\frac{2a_{\max}}{w}$ gestartet wird. Da die LKW-Tour fest gegeben ist und die Drohne die Kunden $i = 1, \dots, n$ an den Orten p_i bedienen muss, lässt

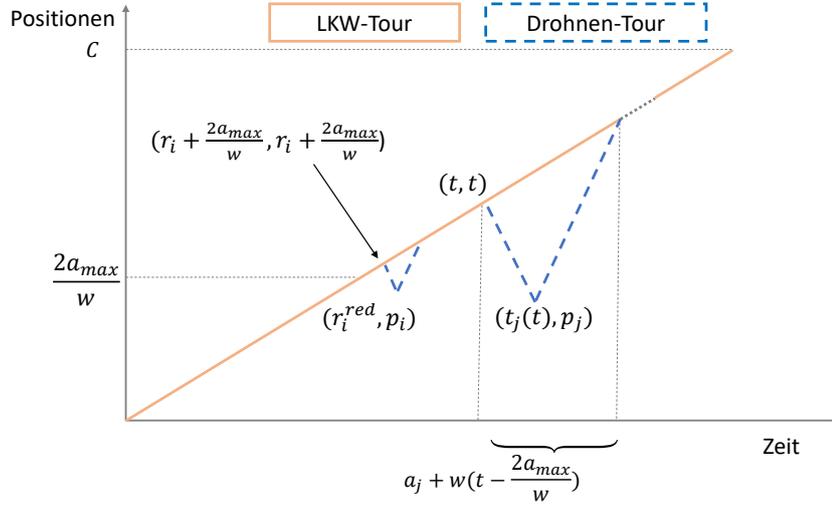


Abbildung 4.12: Teil einer Lösung nach der Reduktion

sich eine Lösung als ein Drohnen-Schedule darstellen.

Sei eine zulässige Lösung für I mit der Reihenfolgefestlegung $(\sigma(1), \dots, \sigma(n))$ und den zugehörigen Fertigstellungszeitpunkten $C_{\sigma(k)}$ für alle $k \in \{1, \dots, n\}$ gegeben. Das bedeutet, dass

$$C_{\sigma(k)} \geq a_{\sigma(k)} + \max \{C_{\sigma(k-1)}w + C_{\sigma(k-1)}, r_{\sigma(k)}w + r_{\sigma(k)}\} \quad (4.30)$$

erfüllt ist. Nun wird gezeigt, dass der Drohnen-Schedule $(\sigma(1), \dots, \sigma(n))$ mit der LKW-Tour $f^T(t) = t$ und den Begegnungszeitpunkten $t_{\sigma(k)}^m = C_{\sigma(k)} + \frac{2a_{\max}}{w}$ eine zulässige Lösung für $\Psi(I)$ ist:

$$\begin{aligned} t_{\sigma(k)}^m &= C_{\sigma(k)} + \frac{2a_{\max}}{w} \stackrel{(4.30)}{\geq} a_{\sigma(k)} + \max \{C_{\sigma(k-1)}(1+w), r_{\sigma(k)}(1+w)\} + \frac{2a_{\max}}{w} \\ &= (2a_{\max} - p_{\sigma(k)}w) + \max \left\{ \left(t_{\sigma(k-1)}^m - \frac{2a_{\max}}{w} \right) (1+w), r_{\sigma(k)}(1+w) \right\} \\ &\quad + \frac{2a_{\max}}{w} \end{aligned} \quad (4.31)$$

Durch Multiplikation von $(1 - \frac{1}{v^D})$ der linken und rechten Seite der Ungleichung (4.31) ergibt sich

$$\begin{aligned} t_{\sigma(k)}^m \left(1 - \frac{1}{v^D}\right) &\geq 2a_{\max} \left(1 + \frac{1}{w}\right) \left(1 - \frac{1}{v^D}\right) - p_{\sigma(k)}w \left(1 - \frac{1}{v^D}\right) \\ &\quad + \max \left\{ \left(1 + \frac{2}{v^D - 1}\right) \left(1 - \frac{1}{v^D}\right) \left(t_{\sigma(k-1)}^m - \frac{2a_{\max}}{w}\right), \right. \\ &\quad \left. r_{\sigma(k)} \left(\frac{2}{v^D - 1} + 1\right) \left(1 - \frac{1}{v^D}\right) \right\} \end{aligned}$$

$$\begin{aligned}
\Leftrightarrow t_{\sigma(k)}^m &\geq \frac{f^T(t_{\sigma(k)}^m)}{v^D} - \frac{2p_{\sigma(k)}}{v^D} + 2a_{\max} \left(1 + \frac{1}{w}\right) \left(1 - \frac{1}{v^D}\right) \\
&\quad + \max \left\{ \left(1 + \frac{1}{v^D}\right) \left(t_{\sigma(k-1)}^m - \frac{2a_{\max}}{w}\right), \left(1 + \frac{1}{v^D}\right) r_{\sigma(k)} \right\} \\
\Leftrightarrow t_{\sigma(k)}^m &\geq \frac{f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}}{v^D} \\
&\quad + \max \left\{ \begin{array}{l} t_{\sigma(k-1)}^m + \frac{f^T(t_{\sigma(k-1)}^m) - p_{\sigma(k)}}{v^D} \\ + 2a_{\max} \left(1 + \frac{1}{w}\right) \left(1 - \frac{1}{v^D}\right) \\ - \frac{2a_{\max}}{w} \left(1 + \frac{1}{v^D}\right), \\ \left(1 + \frac{1}{v^D}\right) r_{\sigma(k)} + 2a_{\max} \left(1 + \frac{1}{w}\right) \left(1 - \frac{1}{v^D}\right) \\ - \frac{p_{\sigma(k)}}{v^D} \end{array} \right\}.
\end{aligned}$$

Somit ist die folgende Bedingung für alle $k \in \{2, \dots, n\}$ erfüllt:

$$t_{\sigma(k)}^m \geq \frac{f^T(t_{\sigma(k)}^m) - p_{\sigma(k)}}{v^D} + \max \left\{ t_{\sigma(k-1)}^m + \frac{f^T(t_{\sigma(k-1)}^m) - p_{\sigma(k)}}{v^D}, r_{\sigma(k)}^{red} \right\}. \quad (4.32)$$

Für $t_{\sigma(k)}^m = C_{\sigma(k)} + \frac{2a_{\max}}{w}$ sind die Abschätzungen (4.31) und (4.32) äquivalent. Mit Lemma 4.6.1 folgt die Zulässigkeit des Drohnen-Schedules mit den angegebenen Begegnungszeitpunkten. Der Zielfunktionswert ist $t_{\sigma(n)}^m = C_{\sigma(n)} + \frac{2a_{\max}}{w}$. Ebenso gilt für eine zulässige Lösung für $\Psi(I)$, dass die Lösung $C_{\sigma(k)} = t_{\sigma(k)}^m - \frac{2a_{\max}}{w}$ mit Zielfunktionswert $C_{\sigma(n)}$ zulässig für I ist. Die Instanz I wird genau dann mit *Ja* beantwortet, wenn $\Psi(I)$ mit *Ja* wird. \square

Satz 4.6.3. *Das Line-TSPwD mit Release Dates und $v^D > v^T$ und $i^* \in \{1, n\}$ mit dem Ziel der Minimierung von $\min\{t_i\}$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Hier wird auf den Beweis von 4.6.2 zurückgegriffen. Allerdings ändert sich dabei der Ort des Kunden $n+1$ sowie sein Release Date, und ein weiterer Kunde $n+2$ wird hinzugefügt.

$$\begin{aligned}
p_{n+1} &= C + \frac{2a_{\max}}{w} + \frac{w}{2+w} = r_{n+1} \\
p_{n+2} &= C + \frac{2a_{\max}}{w} + 1, \quad r_{n+2} = C + \frac{2a_{\max}}{w} + \frac{w}{2+w}
\end{aligned}$$

Die Entscheidungsfrage lautet nun „Existiert eine zulässige Lösung mit einem Zielfunktionswert, der nicht größer als $C + \frac{2a_{\max}}{w} + \frac{w}{2+w}$ ist?“. Aufgrund des Schwellwerts ist es erforderlich, dass die Kunden $n+1$ und $n+2$ zu diesem Zeitpunkt besucht werden. Dies impliziert, dass zum Zeitpunkt $C + \frac{2a_{\max}}{w} + \frac{w}{2+w}$

der LKW und die Drohne am Ort p_{n+1} bzw. p_{n+2} oder umgekehrt am Ort p_{n+2} bzw. p_{n+1} eintrifft. Da

$$\frac{p_{n+2} - p_{i^*}}{v^T} = C + \frac{2a_{\max}}{w} + 1 > C + \frac{2a_{\max}}{w} + \frac{w}{2+w}$$

gilt, kann der LKW diesen Kunden nicht innerhalb der vorgegebenen Zeit bedienen. Das bedeutet, dass der LKW Kunde $n+1$ bedient. Dies setzt voraus, dass der LKW ohne Unterbrechung von Ort p_{i^*} zu p_{n+1} fährt. Die Drohne muss zum Zeitpunkt $C + \frac{2a_{\max}}{w} + \frac{w}{2+w}$ den Ort p_{n+2} erreichen. Dazu muss die Drohne spätestens zum Zeitpunkt $C + \frac{2a_{\max}}{w}$ vom LKW gestartet werden:

$$\begin{aligned} C + \frac{2a_{\max}}{w} + \frac{|f^T(C + \frac{2a_{\max}}{w}) - (C + \frac{2a_{\max}}{w} + 1)|}{v^D} \\ = C + \frac{2a_{\max}}{w} + \frac{|C + \frac{2a_{\max}}{w} - (C + \frac{2a_{\max}}{w} + 1)|}{v^D} \\ = C + \frac{2a_{\max}}{w} + \frac{1}{v^D} = C + \frac{2a_{\max}}{w} + \frac{w}{2+w}. \end{aligned}$$

Der verbleibende Beweis kann analog zum Beweis von 4.6.2 geführt werden. \square

An dieser Stelle kommen wir nochmal auf das Line-TSPwD ohne Zeitfenster zurück. Beim MDRP in (Poikonen & Golden, 2020) wird eine Beschränkung der Drohnenreichweite vorausgesetzt. Diese Beschränkung ist zwar in dem vorliegenden Problem nicht vorausgesetzt, dennoch soll hier erwähnt werden, dass unter dieser Beschränkung bei einem Startkunden $i^* \in \{1, n\}$ das Problem weiterhin polynomiell lösbar ist.

Lemma 4.6.4. *Sei $i^* \in \{1, n\}$ und es seien keine Zeitfenster gegeben. Hat die Drohne im Line-TSPwD eine beschränkte Reichweite und ist sie zusätzlich schneller als der LKW, so lässt sich das Problem in $\mathcal{O}(n^7 \cdot \log n)$ lösen.*

Beweis. Das Problem $1|p_i = a_i - wt_i, r_i|C_{\max}$ ist lösbar in $\mathcal{O}(n^6 \cdot \log n)$. Dies wird in (Cheng & Ding, 1998) gezeigt. Wenn die Drohne und der LKW ganz links oder ganz rechts aller übriger Kunden starten, bewegt sich der LKW strikt in Richtung dieser Kunden. Die Kunden, deren Orte auf der LKW-Tour liegen, werden auch vom LKW bedient. Das Release Date eines Auftrags i kann dann gleich dem frühestmöglichen Startzeitpunkt der Drohne gesetzt werden, ab dem die Drohne aufgrund ihrer begrenzten Reichweite in der Lage ist, den Kunden i zu bedienen. Somit ergibt sich eine Struktur wie in (Cheng & Ding, 1998). \square

4.7 Komplexität mit Deadlines und $v^T > v^D$

Ist die Drohne langsamer als der LKW, so ändert dies tatsächlich nichts an der \mathcal{NP} -Schwere des Problems, falls lediglich Deadlines vorliegen. Dies wird in diesem Abschnitt gezeigt. Dazu wird das 3-Partition-Problem betrachtet.

Definition 4.7.1. Gegeben seien ganze positive Zahlen m und B sowie $3m$ weitere ganze positive Zahlen h_i mit $B/4 < h_i < B/2$ und $\sum_{i \in \{1, \dots, 3m\}} h_i = mB$. Das Problem mit der Entscheidungsfrage „Gibt es eine Partitionierung der Elemente h_i in m Mengen A_1, \dots, A_m , so dass $\sum_{h_i \in A_j} h_i = B$ für alle $j \in \{1, \dots, m\}$?“ heißt 3-Partition Problem. (siehe (Garey & Johnson, 1979))

Satz 4.7.2. ((Garey & Johnson, 1979) Theorem 4.4.) 3-Partition ist \mathcal{NP} -schwer im strengen Sinne.

Hier wird eine 3-Partition Instanz über eine pseudopolynomielle Transformation in eine Instanz des Line-TSPwD mit $v^D < v^T$ und Deadlines überführt. Die Grundidee besteht darin, dass eine LKW-Tour erzwungen wird, bei der der LKW ununterbrochen hin und her pendelt. Dabei befinden sich die Kunden links und rechts näher am Startort, je kleiner die Deadline ist. Die Deadlines sind derart konstruiert, dass der LKW keine andere Wahl hat, als immer auf der einen Seite von p_{i^*} den nächsten noch nicht besuchten Kunden zu bedienen und anschließend auf die andere Seite von p_{i^*} zu fahren, wobei hier der nächste noch nicht besuchte Kunde bedient wird und so weiter. Dadurch werden mögliche Begegnungs- und Trennungsorte festgelegt.

Im Folgenden wird verkürzt vom Kunden p_i gesprochen, wobei p_i den jeweiligen Ort des Kunden bezeichnet. Dies ist möglich, da der Kunde durch seinen Ort eindeutig identifiziert werden kann. Eine Reihenfolgestellung über die Indizes der Kunden ist von nun an nicht mehr vorgesehen. Stattdessen erfolgt diese über den Ort des Kunden. Zur übersichtlicheren Darstellung wird auch statt Ort oder Kunde p_i nur noch von p_i gesprochen.

Die neuen Bezeichnungen vereinfachen die Darstellungen der Instanzen der Komplexitätsbeweise, da dort eine Differenzierung der Kunden in ihren jeweiligen Rollen erfolgt. Dies umfasst beispielsweise Kunden bezüglich der LKW-Tour, für die sich eine bestimmte Reihenfolge ergibt, sowie Drohnenkunden, die einen bestimmten Zweck erfüllen, zum Beispiel zum Einteilen der Drohnen-Tour in getrennte gleich große Zeitsegmente. Ein Beispiel dieser Vorgehensweise ist der Kunde p_i^{right} (siehe Beweis 4.7.3), der durch den LKW bedient wird und zwar als i -ter Kunde, der sich rechts vom Startort p_{i^*} befindet.

Satz 4.7.3. Das Line-TSPwD mit $v^D < v^T$ mit Deadlines und dem Ziel der Minimierung von $\max\{t_i^m, t_i\}$ ist \mathcal{NP} -schwer im strengen Sinne.

Beweis. Sei eine 3-Partition Instanz mit einer ganzen Zahl B und $3m$ weiteren ganzen Zahlen h_i mit $B/4 < h_i < B/2$ und $\sum_{i \in \{1, \dots, 3m\}} h_i = mB$ gegeben, sowie

$$a = \frac{1}{7mB}.$$

Die Line-TSPwD Instanz hat die folgende Struktur:

$$\begin{aligned} v^T &= 4mB \\ v^D &= 1 \\ p_{i^*} &= mB + 1 \\ p_i &= h_i/2 + p_{i^*} + mB && \forall i \in \{1, \dots, 3m\} \\ d_i &= \infty && \forall i \in \{1, \dots, 3m\} \\ p_i^{right} &= p_{i^*} + mB + a \cdot i && \forall i \in \{1, \dots, mB + m\} \\ d_i^{right} &= \frac{1}{4} + (i - 1) + \frac{a \cdot i}{4mB} + \frac{a}{mB} \cdot \frac{i(i - 1)}{2} && \forall i \in \{1, \dots, mB + m\} \\ p_i^{left} &= p_{i^*} - mB - a \cdot i && \forall i \in \{1, \dots, mB + m\} \\ d_i^{left} &= \frac{3}{4} + (i - 1) - \frac{a \cdot i}{4mB} + \frac{a}{mB} \cdot \frac{i(i + 1)}{2} && \forall i \in \{1, \dots, mB + m\} \end{aligned}$$

$$p_i^{en} = p_{i^*} + mB + \frac{1}{2} + \frac{a(iB + i)(4mB + 3) + a(iB + i + 1)(4mB + 1)}{8mB} \quad \forall i \in \{1, \dots, m-1\}$$

$$d_i^{en} = d_{iB+i}^{right} + p_i^{en} - p_{iB+i}^{right} \quad \forall i \in \{1, \dots, m-1\}$$

Die Entscheidungsfrage lautet „Gibt es eine zulässige Lösung, deren Zielfunktionswert nicht größer als d_{mB+m}^{left} ist?“. Hierbei liegen die Kunden an den Positionen p_i^{right} und p_i^{left} für alle $i \in \{1, \dots, mB + m\}$ in der Menge T , sind also durch den LKW zu bedienen. Es werden nun die folgenden Behauptungen bewiesen, die sich auf die konstruierte Instanz $\Psi(I)$ beziehen:

1. Gibt es eine zulässige Lösung für $\Psi(I)$, so werden die Kunden, die der LKW bedient in der Reihenfolge

$$\left(p_{i^*}, p_1^{right}, \dots, p_i^{left}, p_{i+1}^{right}, p_{i+1}^{left}, p_{i+2}^{right}, \dots, p_{mB+m}^{right}, p_{mB+m}^{left} \right)$$

genau zur jeweiligen Deadline besucht.

2. In einer zulässigen Lösung bedient die Drohne alle Kunden p_i mit $i \in \{1, \dots, 3m\}$ und p_j^{en} mit $j \in \{1, \dots, m-1\}$.
3. Falls der Lösungsraum nicht leer ist, so gibt es eine optimale Lösung, bei der die Trennungsorte in der Menge $\{p_i^{right} \mid i \in \{1, \dots, mB + m\}\}$ liegen.
4. Wird die Drohne in einer zulässigen Lösung bei p_i^{right} gestartet, um zu p_k zu fliegen, so kann sie nicht bei p_j^{right} mit $j < i + h_k$ erneut gestartet werden.
5. Wird die Drohne in einer zulässigen Lösung bei p_i^{right} gestartet, um Kunde p_k zu bedienen, so kann die Drohne bei $p_{i+h_k}^{right}$ erneut gestartet werden.
6. Wird die Drohne in einer zulässigen Lösung bei p_{jB+j}^{right} gestartet, um p_j^{en} zu bedienen, so wird p_j^{en} genau zu seiner Deadline erreicht. Die Drohne kann bei p_{jB+j+1}^{right} erneut gestartet werden.
7. Wird die Drohne in einer zulässigen Lösung bei p_k^{right} mit $k < jB + j$ gestartet, um p_j^{en} zu bedienen, so kann die Drohne frühestens beim Kunden p_{k+2}^{right} erneut gestartet werden.

Zunächst wird die erste der obigen Behauptungen gezeigt. Die LKW-Tour wird wegen der Deadlines der Kunden p_i^{left} und p_i^{right} mit $i \in \{1, \dots, mB + m\}$ erzwungen. Als erstes wird also p_1^{right} vom LKW bedient, denn es gilt:

$$\frac{p_1^{right} - p_{i^*}}{v^T} = \frac{p_{i^*} + mB + a - p_{i^*}}{4mB} = \frac{1}{4} + \frac{a}{4mB} \geq d_1^{right} = \frac{1}{4} + \frac{a}{4mB}$$

Darüber hinaus gilt für alle $i, k \in \{1, \dots, mB + m - 1\}$

$$p_{k+1}^{left} < p_k^{left} < p_{i^*} < p_i^{right} < p_{i+1}^{right}.$$

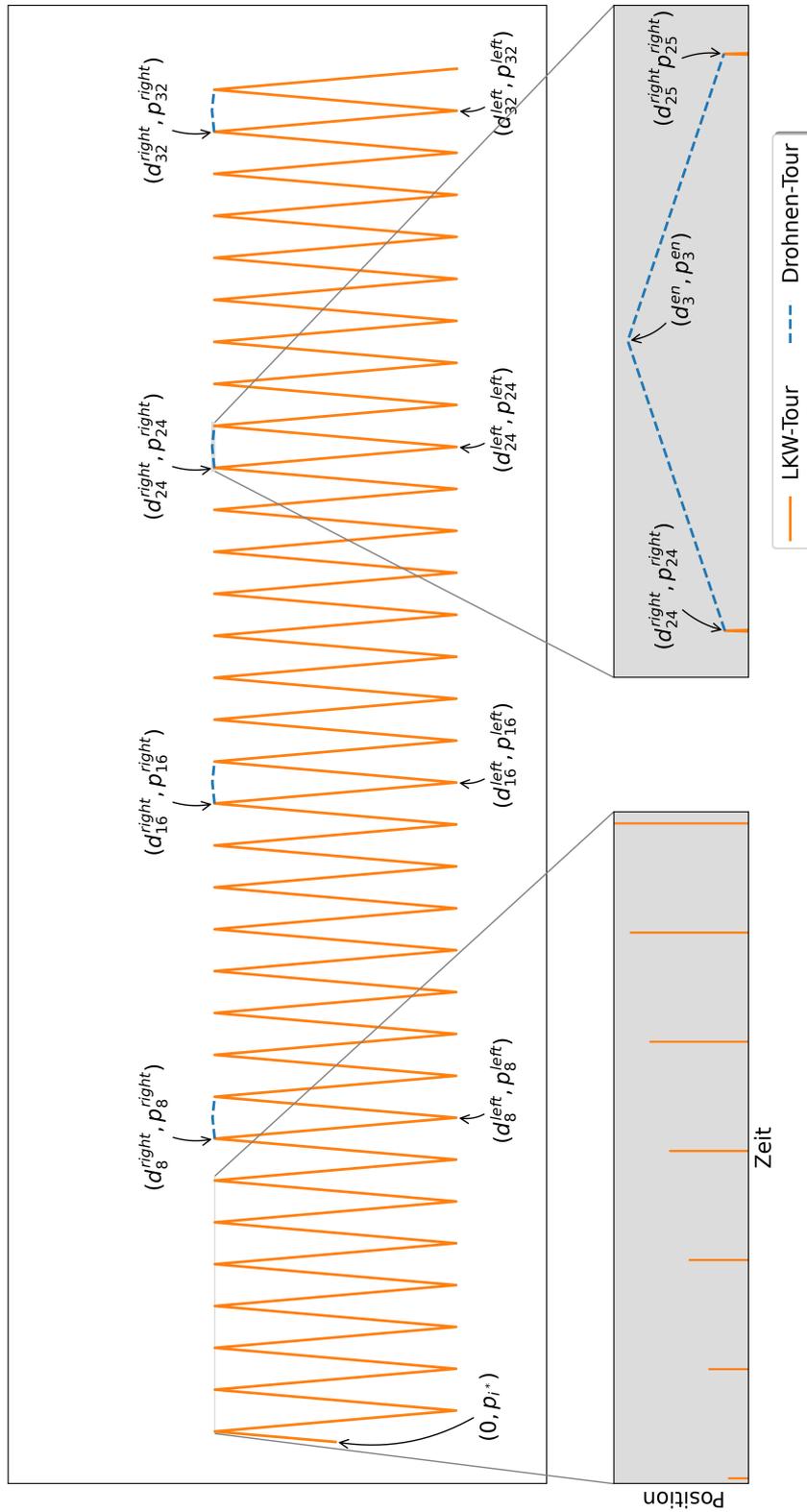


Abbildung 4.13: Line-TSPwD Instanz Reduktion mit $m = 4$ und $B = 7$

Also wurden für $j > i$ die Kunden p_j^{right} noch nicht bedient, solange der LKW nur bis p_i^{right} nach rechts gefahren ist. Analoges gilt für die links von p_i^* . Nachdem p_1^{right} bedient wurde, muss p_1^{left} bedient werden, da gilt

$$\begin{aligned} d_1^{right} + \frac{p_1^{right} - p_1^{left}}{4mB} &= d_1^{right} + \frac{p_{i^*} + mB + a - p_{i^*} + mB + a}{4mB} \\ &= \frac{1}{4} + \frac{a}{4mB} + \frac{2mB + 2a}{4mB} = \frac{3}{4} + \frac{3a}{4mB} = d_1^{left}. \end{aligned}$$

Unter der Voraussetzung, dass p_i^{right} zum Zeitpunkt d_i^{right} bedient wird und der LKW bisher ausschließlich Strecken zwischen den Kunden p_{i-1}^{left} und p_i^{right} zurückgelegt hat, wird Kunde p_i^{left} frühestens zum folgenden Zeitpunkt bedient:

$$\begin{aligned} d_i^{right} + \frac{p_i^{right} - p_i^{left}}{4mB} &= \frac{1}{4} + (i-1) + \frac{a}{mB} \cdot \frac{i(i-1)}{2} + \frac{a \cdot i}{4mB} + \frac{2mB}{4mB} + \frac{2a \cdot i}{4mB} \\ &= \frac{3}{4} + (i-1) + \frac{a}{mB} \cdot \frac{i(i-1)}{2} + \frac{4a \cdot i}{4mB} - \frac{a \cdot i}{4mB} \\ &= \frac{3}{4} + (i-1) + \frac{a}{mB} \cdot \frac{i(i-1) + 2i}{2} - \frac{a \cdot i}{4mB} \\ &= \frac{3}{4} + (i-1) + \frac{a}{mB} \cdot \frac{i(i+1)}{2} - \frac{a \cdot i}{4mB} = d_i^{left}. \end{aligned}$$

Daher muss der LKW, nachdem er den Kunden p_i^{right} bedient hat, zum Kunden p_i^{left} fahren, um diesen rechtzeitig zu erreichen. Ebenso ist der LKW frühestens bei p_{i+1}^{right} zum Zeitpunkt

$$\begin{aligned} d_i^{left} + \frac{p_{i+1}^{right} - p_i^{left}}{4mB} &= \frac{3}{4} + (i-1) + \frac{a}{mB} \cdot \frac{i(i+1)}{2} - \frac{a \cdot i}{4mB} + \frac{a(i+1)}{4mB} + \frac{a \cdot i}{4mB} + \frac{2mB}{4mB} \\ &= \frac{1}{4} + ((i+1) - 1) + \frac{a}{mB} \cdot \frac{((i+1) - 1)(i+1)}{2} + \frac{a(i+1)}{4mB} = d_{i+1}^{right}. \end{aligned}$$

Die resultierende feste LKW-Tour lässt sich anhand der Abbildung 4.13 nachvollziehen. Somit folgt die erste Behauptung.

Eine zulässige Lösung muss gewährleisten, dass alle Kunden bis zum Zeitpunkt d_{mB+m}^{left} bedient werden und die Drohne zu diesem Zeitpunkt auch wieder beim LKW ist. Darüber hinaus muss die Drohne während dieser Zeit alle übrigen Kunden bedienen. Dies sind die Kunden an den Orten p_i^{en} für alle $i \in \{1, \dots, m-1\}$ sowie p_i für alle $i \in \{1, \dots, 3m\}$. Somit gilt auch die zweite Behauptung.

Wird die Drohne zum Zeitpunkt $t \in (d_i^{right}, d_i^{left})$ gestartet, so begegnet sie dem LKW, bevor dieser Kunde p_i^{right} verlassen hat. Dies ist dadurch begründet, dass sich der LKW von p_i^{right} entfernt und die Drohne den LKW nicht einholen kann. In diesem Fall wäre es ebenfalls möglich gewesen, die Drohne bei p_i^{right} zu starten, ohne dass sich der Zielfunktionswert verschlechtert. Wird die Drohne zum Zeitpunkt $t \in (d_i^{left}, d_{i+1}^{right})$ gestartet, so erreicht die Drohne den Ort p_{i+1}^{right} später als der LKW. Daher würde sich der Zielfunktionswert nicht verringern,

falls die Drohne bei p_{i+1}^{right} gestartet wird. Damit ist die dritte Behauptung bewiesen.

Jetzt wird gezeigt, dass die Drohne genau bei $p_{k+h_i}^{right}$ erneut gestartet werden kann, falls sie vorher bei p_k^{right} gestartet wurde, um den Kunden p_i zu bedienen. In diesem Zusammenhang bezeichnet $\Delta^D(i, j, k)$ die Zeit, welche die Drohne benötigt, um von p_i^{right} zu p_k und von p_k zu p_j^{right} zu fliegen.

$$\begin{aligned}\Delta^D(i, j, k) &:= 2p_k - p_i^{right} - p_j^{right} = 2p_{i^*} + 2mB + h_k - p_{i^*} \\ &\quad - mB - a \cdot i - p_{i^*} - mB - a \cdot j = h_k - a(i + j).\end{aligned}$$

Darüber hinaus definiert $\Delta^T(i, j)$ die für die Durchführung der LKW-Tour von p_i^{right} nach p_j^{right} benötigte Zeit.

$$\begin{aligned}\Delta^T(i, j) &:= d_j^{right} - d_i^{right} = (j-1) + \frac{a \cdot j}{4mB} + \frac{a}{mB} \cdot \frac{j(j-1)}{2} \\ &\quad - \left((i-1) + \frac{a \cdot i}{4mB} + \frac{a}{mB} \cdot \frac{i(i-1)}{2} \right) \\ &= (j-i) + \frac{a}{mB} \frac{j(j-1) - i(i-1)}{2} + \frac{a(j-i)}{4mB}.\end{aligned}$$

Gilt die Ungleichung $\Delta^D(i, j, k) > \Delta^T(i, j)$ für $j < i + h_k$, so lässt sich daraus ableiten, dass die Drohne nicht vor $d_{i+h_k}^{right}$ gestartet werden kann. Wegen der Ganzzahligkeit der Parameter gilt:

$$j < i + h_k \Leftrightarrow j + 1 \leq i + h_k \Leftrightarrow j + 1 - i \leq h_k. \quad (4.33)$$

Ferner gilt offensichtlich

$$i, j \leq mB + m \quad (4.34)$$

Nun wird gezeigt, dass obige Ungleichung $\Delta^D(i, j, k) > \Delta^T(i, j)$ erfüllt ist:

$$\begin{aligned}\Delta^D(i, j, k) - \Delta^T(i, j) &= h_k - a(i + j) \\ &\quad - \left((j-i) + \frac{a}{mB} \frac{j(j-1) - i(i-1)}{2} + \frac{a(j-i)}{4mB} \right) \\ &\stackrel{4.33}{\geq} j + 1 - i - j + i - a \cdot i \left(1 - \frac{1}{4mB} \right) - a \cdot j \left(1 + \frac{1}{4mB} \right) \\ &\quad - \frac{a}{mB} \frac{j(j-1) - i(i-1)}{2} \\ &> 1 - a \left(i + j \cdot \left(\frac{4mB+1}{4mB} \right) + \frac{j^2}{2mB} \right) \\ &\stackrel{4.34}{\geq} 1 - a \left((mB+m) + \left(\frac{(mB+m)(4mB+1)}{4mB} \right) + \frac{(mB+m)^2}{2mB} \right) \\ &= 1 - a \left(\frac{4m^2B^2 + 4m^2B + 4m^2B^2 + mB + 4m^2B + m + 2m^2B^2}{4mB} \right) \\ &\quad - a \left(\frac{4m^2B + 2m^2}{4mB} \right) \geq 1 - a \left(\frac{26m^2B^2}{4mB} \right) = 1 - \frac{26m^2B^2}{28m^2B^2} > 0\end{aligned}$$

Zusätzlich wird noch nachgewiesen, dass die Ungleichung $\Delta^D(i, i + h_k, k) \leq \Delta^T(i, i + h_k)$ erfüllt ist. Das bedeutet, dass die Drohne bei $p_{i+h_k}^{right}$ wieder gestartet werden kann:

$$\begin{aligned} & \Delta^T(i, i + h_k) - \Delta^D(i, i + h_k, k) \\ &= i + h_k - i + \frac{a}{mB} \frac{(i + h_k)(i + h_k - 1) - i(i - 1)}{2} + \frac{a(i + h_k - i)}{4mB} \\ & \quad - h_k + a(i + h_k + i) \\ &= \frac{a}{mB} \frac{(i + h_k)(i + h_k - 1) - i(i - 1)}{2} + \frac{a \cdot h_k}{4mB} + a(2i + h_k) > 0 \end{aligned}$$

Also gilt jeweils die vierte und fünfte Behauptung.

In Abbildung 4.14 wird ein Teil einer Lösung der Line-TSPwD-Instanz dargestellt, der zu einer Menge einer 3-Partitions-Instanz korrespondiert. In diesem Fall ist $B = 23$ und es liegen die ganzen Zahlen $h_{\sigma(1)} = 8, h_{\sigma(2)} = 6$ und $h_{\sigma(3)} = 9$ vor. Wird die Drohne bei p_1^{right} gestartet, um den Kunden $p_{\sigma(1)}$ zu bedienen, so kann sie erneut bei $p_{1+h_{\sigma(1)}}^{right} = p_{1+8}^{right}$ gestartet werden. Somit ist es möglich, die Drohne bei p_9^{right} zu starten, um den Kunden $p_{\sigma(2)}$ zu bedienen. Ein erneuter Start ist bei $p_{9+h_{\sigma(2)}}^{right} = p_{15}^{right}$ möglich, um den letzten Kunden dieses Abschnitts zu bedienen.

Damit die Drohne den Kunden am Ort p_j^{en} rechtzeitig bedienen kann, muss sie spätestens bei p_{jB+j}^{right} gestartet werden. Dies lässt sich aus der Gleichung $d_j^{en} = d_{jB+j}^{right} + \frac{p_j^{en} - p_{jB+j}^{right}}{v^D}$ ableiten. Die Drohne kehrt daraufhin umgehend zum LKW zurück. Die Drohne und der LKW begegnen sich exakt bei p_{jB+j+1}^{right} zum Zeitpunkt d_{jB+j+1}^{right} .

$$\begin{aligned} & d_{jB+j}^{right} + \frac{2p_j^{en} - p_{jB+j}^{right} - p_{jB+j+1}^{right}}{v^D} \\ &= \frac{1}{4} + (jB + j - 1) + \frac{a \cdot (jB + j)}{4mB} + \frac{a}{mB} \cdot \frac{(jB + j - 1)(jB + j)}{2} \\ & \quad + 4mB + 2 + 1 + \frac{a(jB + j)(4mB + 3) + a(jB + j + 1)(4mB + 1)}{4mB} \\ & \quad - (2mB + 1 + a(jB + j) + 2mB + 1 + a(jB + j + 1)) \\ &= \frac{1}{4} + (jB + j) + \frac{a(jB + j)}{4mB} + \frac{a}{mB} \cdot \frac{(jB + j - 1)(jB + j)}{2} \\ & \quad + \frac{3a(jB + j) + a(jB + j) + a}{4mB} \\ &= \frac{1}{4} + (jB + j) + \frac{a(jB + j + 1)}{4mB} + \frac{a}{mB} \cdot \frac{(jB + j - 1)(jB + j)}{2} \\ & \quad + \frac{a}{mB} \cdot \frac{2(jB + j)}{2} \\ &= \frac{1}{4} + (jB + j) + \frac{a(jB + j + 1)}{4mB} + \frac{a}{mB} \cdot \frac{((jB + j - 1) + 2)(jB + j)}{2} \\ &= d_{jB+j+1}^{right} \end{aligned}$$

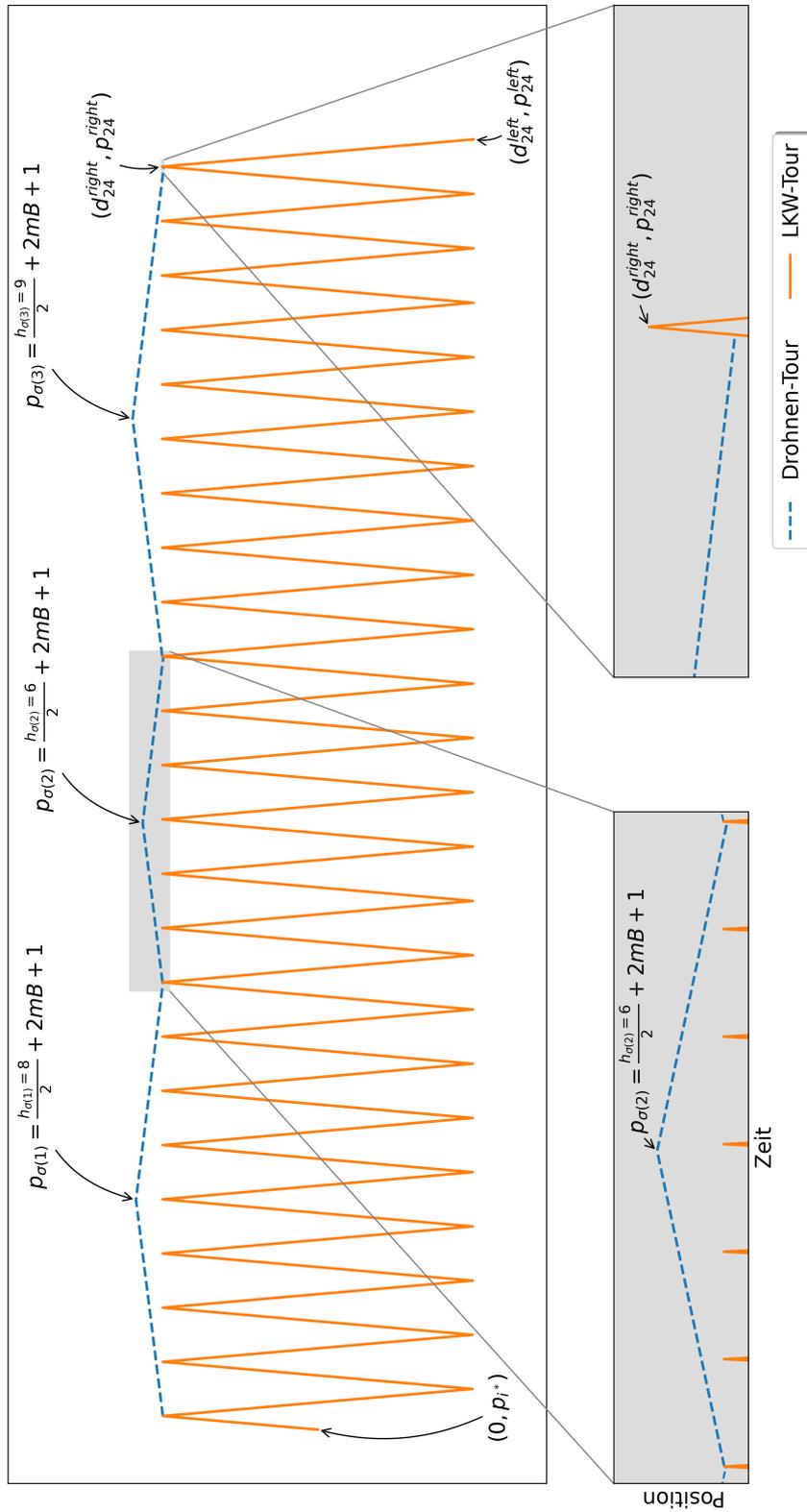


Abbildung 4.14: Eine 3-elementige Menge einer 3-Partitioninstanz mit $B = 23$

Unter der Voraussetzung, dass die Drohne früher gestartet wird, also bei p_k^{right} mit $k < jB + j$, kann sie frühestens bei p_{k+2}^{right} wieder gestartet werden. Die Differenz der Ankunftszeiten von Drohne und LKW bei p_{k+1}^{right} wird in diesem Zusammenhang betrachtet:

$$\begin{aligned}
& \frac{2p_j^{en} - p_k^{right} - p_{k+1}^{right}}{1} - \frac{p_k^{right} + p_{k+1}^{right} - 2p_k^{left}}{4mB} \\
&= 2p_{i^*} + 2mB + 1 + \frac{a(jB + j)(4mB + 3) + a(jB + j + 1)(4mB + 1)}{4mB} \\
&\quad - p_{i^*} - mB - a \cdot k - p_{i^*} - mB - a(k + 1) \\
&\quad - \left(\frac{p_{i^*} + mB + a \cdot k + p_{i^*} + mB + a(k + 1) - 2p_{i^*+2mB+2a_k}}{4mB} \right) \\
&= a(jB + j - k) + \frac{3a(jB + k) - 3a \cdot k}{4mB} + a(jB + j + 1 - (k + 1)) \\
&\quad + \frac{a(jB + j + 1) - a(k + 1)}{4mB} > 0
\end{aligned}$$

Damit sind die Behauptungen 6 und 7 bewiesen.

Sei eine zulässige 3-Partition Instanz mit den m Mengen A_1, \dots, A_m gegeben. Die Drohne wird zu den Zeitpunkten d_{jB+j}^{right} mit $j \in \{1, \dots, m-1\}$ gestartet, um jeweils den Kunden p_j^{en} zu bedienen. Ein erneuter Start der Drohne ist jeweils bei p_{jB+j+1}^{right} möglich. Durch diesen Vorgang resultieren nun m Zeitsegmente, in denen die übrigen Kunden p_j für $j \in \{1, \dots, 3m\}$ durch die Drohne bedient werden können. Werden im i -ten Zeitsegment ab Zeitpunkt $d_{(i-1)+(i-1)B+1}^{right}$ die Kunden $p_{h_1^i}$, $p_{h_2^i}$ und $p_{h_3^i}$ mit $h_1^i, h_2^i, h_3^i \in A_i$ bedient, so erreicht die Drohne rechtzeitig den LKW, um bei p_{i+iB}^{right} gestartet werden zu können. Startet die Drohne bei $p_{(i-1)+(i-1)B+1}^{right}$, um den Kunden $p_{h_1^i}$ zu bedienen, so kann die Drohne wieder bei $p_{(i-1)+(i-1)B+1+h_1^i}^{right}$ gestartet werden, um den Kunden $p_{h_2^i}$ zu bedienen. Die Drohne kann daraufhin wieder zum Zeitpunkt $d_{(i-1)+(i-1)B+1+h_1^i+h_2^i}^{right}$ losfliegen, um den Kunden $p_{h_3^i}$ zu bedienen. Dann ist ein erneutes Starten bei $p_{(i-1)+(i-1)B+1+h_1^i+h_2^i+h_3^i}^{right} = p_{(i-1)+(i-1)B+1+B}^{right} = p_{i+iB}^{right}$ möglich. Diese Lösung ist zulässig für $\Psi(I)$.

Sei nun eine zulässige Lösung für $\Psi(I)$ gegeben, welche die Entscheidungsfrage mit *Ja* beantwortet. In Bezug auf die erste der sieben Aussagen lässt sich festhalten, dass die LKW-Tour eine feste Struktur hat, die der oben beschriebenen entspricht. Es existieren $mB + m$ viele Zeitintervalle $[d_i^{right}, d_{i+1}^{right}]$, was dementsprechend $mB + m$ mögliche Startzeitpunkte der Drohne zur Folge hat (zweite Aussage). Um den Kunden p_k zu bedienen, sind h_k -viele solcher Zeitintervalle notwendig (4. und 5. Aussage), also insgesamt $\sum h_k = mB$ viele. Um den Kunden p_j^{en} zu bedienen, ist ebenfalls mindestens ein Zeitintervall erforderlich. Es gibt insgesamt m solcher Kunden. Sollte die Drohne nicht bei p_{jB+j}^{right} gestartet werden, um den Kunden p_j^{en} zu bedienen, so ist entweder eine rechtzeitige Ankunft bei diesem Kunden nicht mehr gewährleistet, oder zwei Zeitintervalle sind erforderlich. Um zu vermeiden, dass die Anzahl der $mB + m$ Zeitintervalle überschritten wird, muss die Drohne bei p_{jB+j}^{right} gestartet werden, um den Kunden p_j^{en} zu bedienen. Also ergeben sich m separate Zeitabschnitte

mit B Zeitintervallen. Innerhalb eines Zeitsegments können maximal drei Kunden durch die Drohne bedient werden, da $h_i > B/4$ gilt. Für die drei Kunden h_1, h_2 und h_3 muss die Bedingung $h_1 + h_2 + h_3 = B$ erfüllt sein. \square

Satz 4.7.4. *Das Line-TSPwD mit $v^D < v^T$ und Deadlines mit dem Ziel der Minimierung von $\max\{t_i\}$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Eine 3-Partition-Instanz I wird in eine Line-TSPwD-Instanz überführt. Dazu wird $\Psi(I)$ aus dem Beweis von Satz 4.7.3 benötigt, wobei ein Kunde p_{last} hinzugefügt wird und sich der Ort und die Deadline von Kunde p_{mB+m}^{left} ändert:

$$\begin{aligned} p_{last} &= 6(mB + m + 1)^2 & d_{last} &= d_{mB+m}^{right} + \frac{p_{last} - p_{mB+m}^{right}}{v^D} \\ p_{mB+m}^{left} &= p_{mB+m}^{right} - v^T \left(d_{last} - d_{mB+m}^{right} \right) & d_{mB+m}^{left} &= d_{last}. \end{aligned}$$

Die Entscheidungsfrage lautet „Lässt sich eine zulässige Lösung mit einem Zielfunktionswert finden, der nicht größer als d_{last} ist?“.

Wie in 4.7.3 ist es auch hier erforderlich, dass die LKW-Tour der ersten Aussage im Beweis von 4.7.3 genügt. Dazu muss noch nachgewiesen werden, dass der neue Kunde p_{mB+m}^{left} rechtzeitig bedient wird. Dabei ist zu beachten, dass die Ungleichung $p_{mB+m}^{left} < p_{mB+m-1}^{left}$ erfüllt ist, also der LKW bis zum Zeitpunkt d_{mB+m}^{right} diesen Kunden noch nicht bedient haben kann:

$$\begin{aligned} p_{mB+m}^{left} &= p_{mB+m}^{right} - v^T \left(d_{last} - d_{mB+m}^{right} \right) \\ &= p_{mB+m}^{right} - 4mB \left(d_{mB+m}^{right} + p_{last} - p_{mB+m}^{right} - d_{mB+m}^{right} \right) \\ &= (1 + 4mB) p_{mB+m}^{right} - 24mB(mB + m + 1)^2 \\ &< (1 + 4mB)(2mB + 2) - 24mB(mB + m + 1)^2 \\ &= (4mB + 1)(2mB + 2) - (12mB + 12m + 12)(2mB + 2m + 2)mB \\ &< 0 < p_{mB+m-1}^{left}. \end{aligned}$$

Der LKW entfernt sich ab dem Zeitpunkt d_{mB+m}^{right} strikt von den Kunden, die durch die Drohne bedient wurden. Da die Drohne den LKW nicht einholen kann, wird ab diesem Zeitpunkt nur noch ein Kunde von der Drohne bedient. Die Reisezeit, welche die Drohne benötigt, um von p_i^{right} zu p_{last} und von p_{last} zu p_j^{right} zu fliegen, lässt sich wie folgt abschätzen:

$$\begin{aligned} \frac{p_{last} - p_i^{right}}{v^D} + \frac{p_{last} - p_j^{right}}{v^D} &> 2p_{last} - 2(2mB + 2) > 8(mB + m + 1)^2 \\ &> \frac{1}{4} + (mB + m - 1) + \frac{mB + m}{28m^2B^2} + \frac{(mB + m)(mB + m - 1)}{28m^2B^2} \\ &= d_{mB+m}^{right}. \end{aligned}$$

Also ist p_{last} der letzte von der Drohne bediente Kunde. Wird die Drohne bei p_{mB+m}^{right} gestartet, um den Kunden p_{last} zu bedienen, so ist die Ankunftszeit der Drohne bei diesem Kunden noch rechtzeitig zum Zeitpunkt d_{last} . Der restliche Beweis kann analog zum Beweis von Satz 4.7.3 geführt werden. \square

Ist $i^* \in \{1, n\}$, sind lediglich Deadlines vorhanden und ist die Drohne langsamer als der LKW, so lässt sich das Problem trivial lösen. Der LKW fährt von i^* konstant zum anderen Ende der Linie und bedient direkt beim Vorbeifahren den jeweiligen Kunden. Die Drohne bedient keinen der Kunden. Diese Lösung ist zulässig, sofern $\frac{|p_{i^*} - p_i|}{v^T} \leq d_i$ für alle Kunden i erfüllt ist.

4.8 Komplexität mit Release Dates und $v^T > v^D$

Unter der Voraussetzung, dass lediglich Release Dates, jedoch keine Deadlines gegeben sind, stellt sich das Problem weiterhin als \mathcal{NP} -schwer im strengen Sinne dar. Der Beweis in diesem Abschnitt orientiert sich an der Idee des Beweises von Satz 4.7.3.

Satz 4.8.1. *Das Line-TSPwD mit $v^D < v^T$ und Release Dates sowie $i^* \in \{1, n\}$ mit dem Ziel der Minimierung von $\max\{t_i^m, t_i\}$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Sei eine 3-Partition Instanz I mit einer ganzen Zahl B und $3m$ weiteren ganzen Zahlen h_i mit $B/4 < h_i < B/2$ und $\sum_{i \in \{1, \dots, 3m\}} h_i = mB$ gegeben. Außerdem seien

$$a = \frac{1}{9mB(mB+m)} \text{ und } a_i = a \cdot (mB + m - i).$$

Die Line-TSPwD Instanz $\Psi(I)$ hat die folgende Struktur:

$$v^T = 4mB$$

$$v^D = 1$$

$$p_{i^*} = 0$$

$$p_i = h_i/2 + 4mB \quad \forall i \in \{1, \dots, 3m\}$$

$$r_i = 0 \quad \forall i \in \{1, \dots, 3m\}$$

$$p_i^{right} = 4mB + a_i \quad \forall i \in \{1, \dots, mB + m\}$$

$$r_i^{right} = i + \frac{a_i}{4mB} + \frac{a(i-1)(2mB+2m-i)}{2mB} \quad \forall i \in \{1, \dots, mB + m\}$$

$$p_i^{left} = 2mB - a_i \quad \forall i \in \{1, \dots, mB + m\}$$

$$r_i^{left} = i + \frac{1}{2} + \frac{3a_i}{4mB} + \frac{a(i-1)(2mB+2m-i)}{2mB} \quad \forall i \in \{1, \dots, mB + m\}$$

$$p_i^{en} = 4mB + \frac{1}{2} + \left(\frac{4mB+3}{8mB}\right)a_{iB+i} + \left(\frac{4mB+1}{8mB}\right)a_{iB+i+1} \quad \forall i \in \{1, \dots, m-1\}$$

$$r_i^{en} = r_{iB+i}^{right} + p_i^{en} - p_{iB+i}^{right} \quad \forall i \in \{1, \dots, m-1\}$$

Die Entscheidungsfrage lautet „Gibt es eine zulässige Lösung, deren Zielfunktionswert nicht größer als r_{mB+m}^{left} ist?“. Die Kunden p_i^{right} und p_i^{left} für alle $i \in \{1, \dots, mB + m\}$ befinden sich in der Menge T und können somit ausschließlich durch den LKW bedient werden.

Wie beim Beweis des Satzes 4.7.3 wird nun gezeigt, dass die folgenden Aussagen richtig sind:

1. Liegt eine zulässige Lösung für $\Psi(I)$ vor, die mit *Ja* beantwortet wird, so werden die Kunden in der Reihenfolge

$$\left(p_{i^*}, p_1^{right}, \dots, p_i^{left}, p_{i+1}^{right}, p_{i+1}^{left}, p_{i+2}^{right}, \dots, p_{mB+m}^{right}, p_{mB+m}^{left} \right)$$

vom LKW genau zum jeweiligen Release Date bedient.

2. In einer zulässigen Lösung, die mit *Ja* beantwortet wird, bedient die Drohne alle Kunden p_i mit $i \in \{1, \dots, 3m\}$ und p_j^{en} mit $j \in \{1, \dots, m\}$.
3. Existiert eine Lösung mit einer *Ja*-Antwort, so gibt es eine Lösung, bei der die Trennungsorte in der Menge $\{p_i^{right} \mid i \in \{1, \dots, mB + m\}\}$ liegen.
4. Bei Vorliegen einer LKW-Tour gemäß der ersten Behauptung und einem Start der Drohne bei p_i^{right} mit anschließendem Flug zu p_k ist ein erneuter Start bei p_j^{right} mit $j < i + h_k$ nicht möglich.
5. Sei eine LKW-Tour gemäß der ersten Behauptung gegeben. Startet die Drohne am Ort p_i^{right} , um den Kunden p_k zu bedienen, so kann diese bei $p_{i+h_k}^{right}$ erneut gestartet werden.
6. Bei einer LKW-Tour gemäß der ersten Behauptung und bei einem Start der Drohne bei p_{jB+j}^{right} , um p_j^{en} zu bedienen, wird p_j^{en} exakt zum Release Date erreicht. Die Drohne kann anschließend wieder bei p_{jB+j+1}^{right} gestartet werden.
7. Im Falle einer LKW-Tour gemäß der ersten Behauptung und bei einem Start der Drohne bei p_k^{right} mit $k > jB + j$, um den Kunden p_j^{en} zu bedienen, kann die Drohne frühestens bei p_{k+2}^{right} erneut gestartet werden.

Zunächst wird die erste dieser Behauptungen gezeigt. Da $r_{m(B+1)}^{left}$ den Zeitpunkt markiert, zu dem die LKW-Tour beendet sein muss, befindet sich der LKW zu diesem Zeitpunkt bei $p_{m(B+1)}^{left}$. Es ist somit erforderlich, dass sich der LKW zum Zeitpunkt r_{mB+m-1}^{left} beim Kunden p_{mB+m-1}^{left} befindet, damit eine mit *Ja* beantwortbare Lösung vorliegt:

$$\begin{aligned} r_{mB+m}^{right} - \frac{p_{mB+m}^{right} - p_{mB+m-1}^{left}}{4mB} &= (mB + m) + \frac{a(mB + m - mB - m)}{4mB} \\ &+ \frac{a(mB + m - 1)(2mB + 2m - mB - m)}{2mB} \\ &- \frac{4mB + a(mB + m - mB - m) - 2mB + a(mB + m - mB - m + 1)}{4mB} \\ &= (mB + m - 1) + \frac{1}{2} + \frac{a(mB + m - 1)(mB + m)}{2mB} - \frac{a}{4mB} \\ &= (mB + m - 1) + \frac{1}{2} + \frac{3a}{4mB} + \frac{2a(mB + m - 1)(mB + m) - 4a}{4mB} \\ &= (mB + m - 1) + \frac{1}{2} + \frac{3a(mB + m - mB - m + 1)}{4mB} \\ &+ \frac{a\left((mB + m)^2 - (mB + m) - 2\right)}{2mB} \end{aligned}$$

$$\begin{aligned}
&= (mB + m - 1) + \frac{1}{2} + \frac{3a(mB + m - mB - m + 1)}{4mB} \\
&\quad + \frac{a((mB + m - 2)(mB + m + 1))}{2mB} \\
&= (mB + m - 1) + \frac{1}{2} + \frac{3a(mB + m - mB - m + 1)}{4mB} \\
&\quad + \frac{a(mB + m - 2)(2mB + 2m - (mB + m - 1))}{2mB} = r_{mB+m-1}^{left}
\end{aligned}$$

Unter der Annahme, dass sich der LKW ab dem Zeitpunkt r_{k+1}^{right} innerhalb des Intervalls $[p_{k+1}^{right}, p_{k+1}^{left}]$ bewegt und zu den Zeitpunkten r_j^{right} bzw. r_j^{left} die Kunden p_j^{right} bzw. p_j^{left} mit $j > k$ bedient, so muss der Kunde p_k^{left} vorher bedient werden, da der LKW diesen Ort später nicht mehr erreicht. Daher muss für den Fertigstellungszeitpunkt $t_{(k, left)}$ des Kunden p_k^{left} gelten:

$$\begin{aligned}
t_{(k, left)} &\leq r_{k+1}^{right} - \frac{p_{k+1}^{right} - p_k^{left}}{4mB} = (k+1) + \frac{a(mB + m - k - 1)}{4mB} \\
&\quad + \frac{a \cdot k(2mB + 2m - k - 1)}{2mB} \\
&\quad - \left(\frac{4mB + a(mB + m - k - 1) - 2mB + a(mB + m - k)}{4mB} \right) \\
&= k + \frac{1}{2} + \frac{a(mB + m - k)}{4mB} - \frac{a}{4mB} + \frac{a \cdot k(2mB + 2m - k)}{2mB} - \frac{a \cdot k}{2mB} \\
&\quad - \frac{2a(mB + m - k)}{4mB} + \frac{a}{4mB} \\
&= k + \frac{1}{2} + \frac{3a(mB + m - k)}{4mB} + \frac{a(k-1)(2mB + 2m - k)}{2mB} \\
&\quad + \frac{a(2mB + 2m - k)}{2mB} - \frac{a \cdot k}{2mB} - \frac{4a(mB + m - k)}{4mB} \\
&= k + \frac{1}{2} + \frac{3a(mB + m - k)}{4mB} + \frac{a(k-1)(2mB + 2m - k)}{2mB} \\
&\quad + \frac{a(4mB + 4m - 2k - 2k - 4mB - 4m + 4k)}{4mB} \\
&= k + \frac{1}{2} + \frac{3a(mB + m - k)}{4mB} + \frac{a(k-1)(2mB + 2m - k)}{2mB} = r_k^{left}
\end{aligned}$$

Folglich muss dieser Kunde exakt zum Zeitpunkt r_k^{left} aufgesucht werden. Analog lässt sich für den Fertigstellungszeitpunkt $t_{(k, right)}$ des Kunden p_k^{right} Folgendes ableiten:

$$\begin{aligned}
t_{(k, right)} &\leq r_k^{left} - \frac{p_k^{right} - p_k^{left}}{4mB} = k + \frac{1}{2} + \frac{3a(mB + m - k)}{4mB} \\
&\quad + \frac{a(k-1)(2mB + 2m - k)}{2mB} \\
&\quad - \left(\frac{4mB + a(mB + m - k) - 2mB + a(mB + m - k)}{4mB} \right) \\
&= k + \frac{a(mB + m - k)}{4mB} + \frac{a(k-1)(2mB + 2m - k)}{2mB} = r_k^{right}
\end{aligned}$$

Somit bedient der LKW seine Kunden in der Reihenfolge

$$\left(p_{i^*}, p_1^{right}, p_1^{left}, \dots, p_i^{right}, p_i^{left}, \dots, p_{m(B+1)}^{left} \right),$$

wobei der LKW immer bei diesen Kunden sofort wendet und niemals wartet. Damit folgt die erste Behauptung. Die Struktur der LKW-Tour ist in Abbildung 4.15 zu erkennen.

Für alle $i \in \{1, \dots, 3m\}$, $j \in \{1, \dots, mB + m\}$ und $k \in \{1, \dots, m\}$ gilt

$$p_i = \frac{h_i}{2} + 4mB > \frac{B}{8} + 4mB > \frac{(mB + m - 1)}{9mB(mB + m)} + 4mB = p_1^{right} \geq p_j^{right}$$

und

$$p_k^{en} > 4mB + \frac{1}{2} > 4mB + \frac{mB + m - 1}{9mB(mB + m)} = p_1^{right} \geq p_j^{right}. \quad (4.35)$$

Daher müssen die Kunden p_k^{en} für alle $k \in \{1, \dots, m - 1\}$ und p_i für alle $i \in \{1, \dots, 3m\}$ durch die Drohne bedient werden. Es folgt also die zweite Behauptung.

Aus dem Beweis des Satzes 4.7.3 kann die Begründung der dritten Behauptung entnommen werden.

Seien $\Delta^D(i, j, k)$ und $\Delta^T(i, j)$ wie im Beweis von Satz 4.7.3 definiert. Im Folgenden wird gezeigt, dass die Drohne nicht bei p_j^{right} für $j < k + h$ erneut gestartet werden kann, falls sie bei p_k^{right} gestartet wird, um Kunde p_i zu bedienen. Wegen der Ganzzahligkeit von h_i, j und k gilt $j - k + 1 \leq h_i$. Außerdem gelten die Beschränkungen $i, j \leq mB + m$.

$$\begin{aligned} \Delta^D(k, j, i) - \Delta^T(k, j) &= 2p_i - p_k^{right} - p_j^{right} - (r_j^{right} - r_k^{right}) \\ &= h_i - a_k - a_j - \left((j - k) \right. \\ &\quad \left. + \frac{a(k - j)}{4mB} + \frac{a(j - k)(2mB + 2m)}{2mB} + \frac{a((k - 1)k - (j - 1)j)}{2mB} \right) \\ &\geq j - k + 1 - a(2mB + 2m - k - j) + \frac{a(k - j)(2mB + 2m)}{2mB} \\ &\quad - j + k + \frac{a(j - k)}{4mB} + \frac{2a(j(j - 1) - k(k - 1))}{4mB} \\ &> 1 - a(2mB + 2m - k - j) - \frac{a(j - k)(2mB + 2m)}{2mB} \\ &= 1 - \left(\frac{a(2mB(2mB + 2m - k - k) + (j - k)(2mB + 2m))}{2mB} \right) \\ &\geq 1 - \left(\frac{2mB(2mB + 2m) + (mB + m)(2mB + 2m)}{9mB(mB + m)2mB} \right) \\ &= 1 - \left(\frac{6m^2B^2 + 8m^2B + 2m^2}{18m^2B^2(mB + m)} \right) > 0 \end{aligned}$$

Also ist die 4. Behauptung ebenfalls richtig.

Um einen Schwellwert von r_{mB+m}^{left} bezüglich $\Psi(I)$ zu erreichen, muss die Drohne

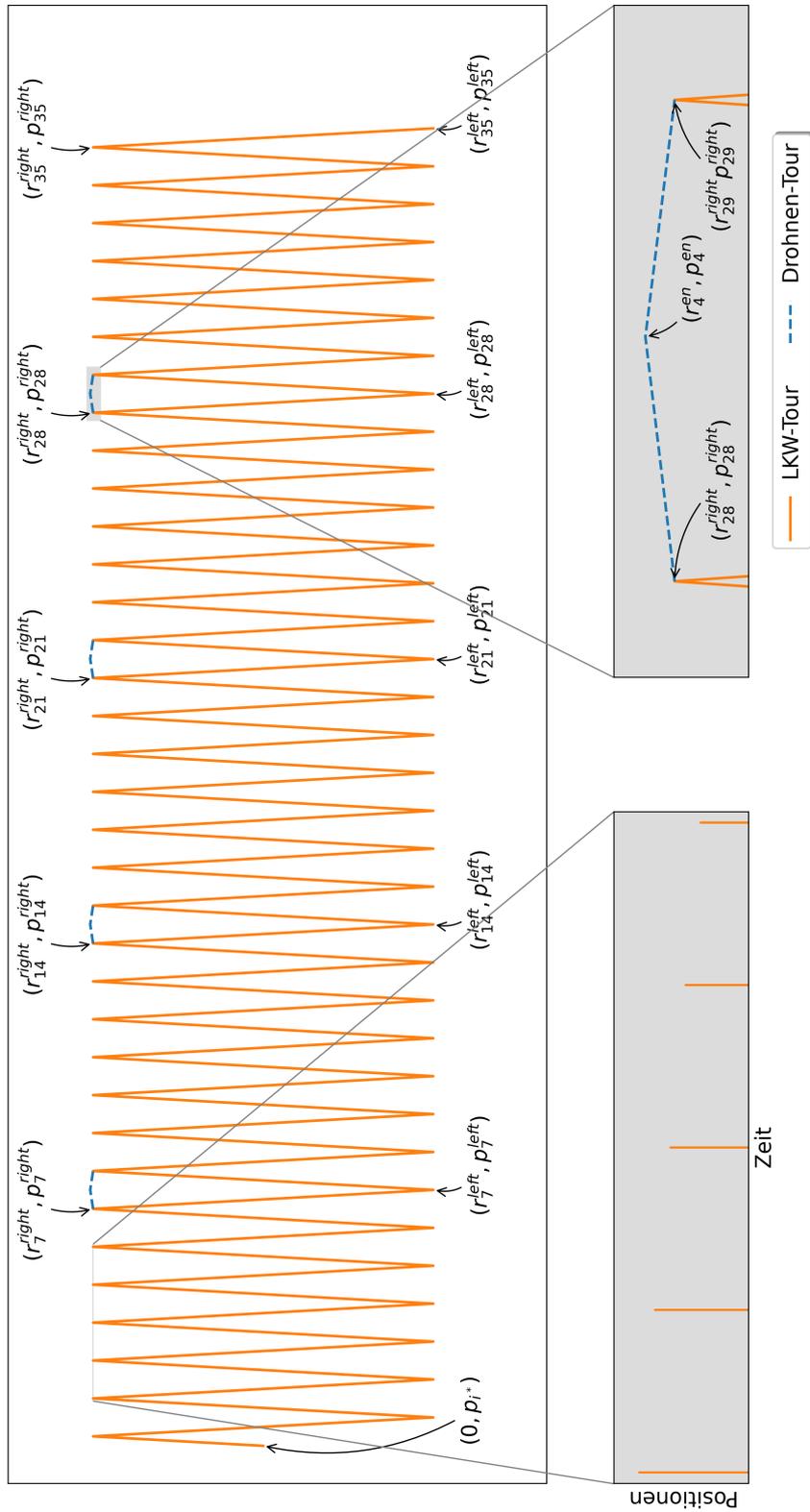


Abbildung 4.15: Line-TSPwD Instanz Reduktion mit $m=5$ und $B = 6$

spätestens zum Zeitpunkt r_{mB+m}^{right} bei p_{mB+m}^{right} sein, nachdem sie ihren letzten Kunden bedient hat. Würde die Drohne bei p_k^{right} mit $k \geq mB+m-h_i$ gestartet werden, um den Kunden p_i zu bedienen, so kann sie nach Behauptung 4 nicht wieder bei p_j^{right} erneut gestartet werden, wobei $j < h_i+k$ ist. Jedoch gilt $h_i+k \geq mB+m$. Daher wird Folgendes vorausgesetzt

$$h_i+k \leq mB+m. \quad (4.36)$$

Es wird nun gezeigt, dass die Reisezeit der Drohne, um von p_k^{right} zu p_i und von p_i zu $p_{k+h_i}^{right}$ zu fliegen, geringer ist, als die Reisezeit des LKWs unter Berücksichtigung seiner festen Tour.

$$\begin{aligned} \Delta^T(k, k+h_i) - \Delta^D(k, k+h_i, i) &= (k+h_i-k) + \frac{a(k-h_i-k)}{4mB} \\ &+ \frac{a(k+h_i-1)(2mB+2m-(k+h_i))}{2mB} - \frac{a(k-1)(2mB+2m-k)}{2mB} \\ &- h_i + a(mB+m-k) + a(mB+m-(h_i+k)) \\ &= -\frac{a \cdot h_i}{4mB} + \frac{a(k-1)(2mB+2m-(k+h_i))}{2mB} + \frac{a \cdot h_i(2mB+2m-(k+h_i))}{2mB} \\ &- \frac{a(k-1)(2mB+2m-k)}{2mB} + a(mB+m-k) + a(mB+m-(h_i+k)) \\ &= -\frac{a \cdot h_i}{4mB} + \frac{a(k-1)(2mB+2m-(k+h_i)-2mB-2m+k)}{2mB} \\ &+ \frac{a \cdot h_i(2mB+2m-(k+h_i))}{2mB} + a(mB+m-k) + a(mB+m-(h_i+k)) \\ &= -\frac{a \cdot h_i}{4mB} + \frac{a(k-1)(-h_i)}{2mB} + \frac{a \cdot h_i(2mB+2m-(k+h_i))}{2mB} \\ &+ \frac{2mB \cdot a(2mB+2m) - 2mB \cdot a \cdot h_i - 4mB \cdot a \cdot k}{2mB} \\ &\stackrel{4.36}{\geq} -\frac{a \cdot h_i}{4mB} + \frac{a(mB+m-h_i)(-h_i)}{2mB} + \frac{a \cdot h_i(2mB+2m-(mB+m))}{2mB} \\ &+ \frac{2mB \cdot a(2mB+2m) - 2mB \cdot a(h_i+k+k)}{2mB} \\ &= -\frac{a \cdot h_i}{4mB} + \frac{a \cdot h_i(mB+m-mB-m+h_i)}{2mB} + a(2mB+2m-((h_i+k)+k)) \\ &> -\frac{a \cdot h_i}{4mB} + \frac{a \cdot h_i^2}{2mB} = \frac{2ah_i^2 - ah_i}{4mB} > 0 \end{aligned}$$

Somit folgt die 5. Behauptung.

Bei einem Start der Drohne am Ort p_{jB+j}^{right} , um zum Kunden p_j^{en} zu fliegen, kann die Drohne erneut bei p_{jB+j+1}^{right} gestartet werden:

$$\begin{aligned} r_{jB+j}^{right} &+ \frac{2p_j^{en} - p_{jB+j}^{right} - p_{jB+j+1}^{right}}{v^D} \\ &= (jB+j) + \frac{a_{jB+j}}{4mB} + \frac{a(jB+j-1)(2mB+2m-(jB+j))}{2mB} \\ &+ 2 \left(4mB + \frac{1}{2} + \left(\frac{4mB+3}{8mB} \right) a_{jB+j} + \left(\frac{4mB+1}{8mB} \right) a_{jB+j+1} \right) \end{aligned}$$

$$\begin{aligned}
& -4mB - a_{jB+j} - 4mB - a_{jB+j+1} \\
= & (jB + j + 1) + \frac{a(mB + m - (jB + j))}{4mB} + \frac{3}{4mB} \cdot a(mB + m - (jB + j)) \\
& + \frac{a(jB + j - 1)(2mB + 2m - (jB + j))}{2mB} \\
& + \frac{1}{4mB} \cdot a(mB + m - (jB + j + 1)) \\
= & (jB + j + 1) + \frac{a(mB + m - (jB + j + 1))}{4mB} \\
& + \frac{a(jB + j - 1)(2mB + 2m - (jB + j))}{2mB} \\
& + \frac{a(mB + m - (jB + j))}{4mB} + \frac{3a(mB + m - (jB + j))}{4mB} \\
= & (jB + j + 1) + \frac{a(mB + m - (jB + j + 1))}{4mB} \\
& + \frac{a(jB + j - 1)(2mB + 2m - (jB + j))}{2mB} + \frac{a(2mB + 2m - (jB + j))}{2mB} \\
& - \frac{a(jB + j)}{2mB} \\
= & (jB + j + 1) + \frac{a(mB + m - (jB + j + 1))}{4mB} \\
& + \frac{a(jB + j)(2mB + 2m - (jB + j))}{2mB} - \frac{a(jB + j)}{2mB} \\
= & (jB + j + 1) + \frac{a(mB + m - (jB + j + 1))}{4mB} \\
& + \frac{a(jB + j)(2mB + 2m - (jB + j + 1))}{2mB} = r_{jB+j+1}^{right}
\end{aligned}$$

Somit ist auch die 6. Behauptung richtig.

Um die letzte Behauptung zu beweisen, wird nachgewiesen, dass die Reisezeit der Drohne der Tour $(p_k^{right}, p_j^{en}, p_{k+1}^{right})$ größer ist, als die Reisezeit des LKWs der Tour $(p_k^{right}, p_k^{left}, p_{k+1}^{right})$ für $k > jB + j$.

$$\begin{aligned}
& 2p_j^{en} - p_k^{right} - p_{k+1}^{right} - \frac{p_k^{right} + p_{k+1}^{right} - 2p_k^{left}}{4mB} \\
= & 2 \left(4mB + \frac{1}{2} + \left(\frac{4mB + 3}{8mB} \right) a_{jB+j} + \left(\frac{4mB + 1}{8mB} \right) a_{jB+j+1} \right) \\
& - \left(p_k^{right} \left(1 + \frac{1}{4mB} \right) + p_{k+1}^{right} \left(1 + \frac{1}{4mB} \right) - 2p_k^{left} \left(\frac{1}{4mB} \right) \right) \\
= & 2 \left(4mB + \frac{1}{2} \right) + \left(\frac{4mB + 3}{4mB} \right) a_{jB+j} + \left(\frac{4mB + 1}{4mB} \right) a_{jB+j+1} \\
& - \left(2 \cdot 4mB \left(1 + \frac{1}{4mB} \right) + \frac{4mB}{4mB} + \left(\frac{4mB + 3}{4mB} \right) a_k + \left(\frac{4mB + 1}{4mB} \right) a_{k+1} \right) \\
= & 8mB - 8mB + 1 - \frac{8mB}{4mB} + \frac{4mB}{4mB}
\end{aligned}$$

$$\begin{aligned}
& + \left(\frac{4mB+3}{4mB} \right) \cdot a(mB+m-(jB+j)-mB-m+k) \\
& + \left(\frac{4mB+3}{4mB} \right) \cdot a(mB+m-(jB+j+1)-mB-m+k+1) \\
& = \left(\frac{4mB+3}{4mB} \right) \cdot a(k-(jB+j)) + \left(\frac{4mB+1}{4mB} \right) \cdot a(k+1-(jB+j+1)) > 0
\end{aligned}$$

In der Beweisführung des Theorems 4.7.3 liegt eine vergleichbare Struktur vor. An dieser Stelle wird auf diesen Beweis verwiesen. \square

Satz 4.8.2. *Das Line-TSPwD mit $v^D < v^T$ und Release Dates sowie $i^* \in \{1, n\}$ mit dem Ziel der Minimierung von $\max\{t_i\}$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Auch hier wird auf die Grundidee des Beweises von Satz 4.8.1 verwiesen. Zur Instanz $\Psi(I)$ wird ein weiterer Kunde p_{last} mit der folgenden Position und dem Release Date hinzugefügt:

$$p_{last} = d_{mB+m}^{right} - d_{mB+m}^{left} + p_{mB+m}^{right} \quad r_{last} = r_{mB+m}^{left}.$$

Da der LKW zum Zeitpunkt r_{mB+m}^{left} bei p_{mB+m}^{left} sein muss, befindet sich die Drohne zu diesem Zeitpunkt bei p_{last} . Bei einem Start der Drohne am Ort p_{mB+m}^{right} erreicht sie den Kunden p_{last} exakt zum Zeitpunkt r_{mB+m}^{left} . \square

4.9 Komplexität des diskreten Falls

Bisher war das Auflesen und Starten der Drohne an jedem Ort erlaubt. Jetzt wird allerdings vorausgesetzt, dass dies nur noch an vorgegebenen Positionen möglich ist. In dem Fall, dass der LKW schneller ist als die Drohne, lässt sich aus 4.2.3 ableiten, dass das Problem polynomiell lösbar ist, sofern keine Zeitfenster vorausgesetzt sind.

Auch wenn entweder nur Deadlines oder nur Release Dates vorhanden sind und die Drohne eine geringere Geschwindigkeit als der LKW hat, können die Ergebnisse des kontinuierlichen Falls genutzt werden. In den Beweisen von 4.7.3 und 4.8.1, welche die \mathcal{NP} -Schwere im kontinuierlichen Fall zeigen, wurde eine dominante Lösung gefunden, bei der die Drohne bei Kundenorten gestartet und aufgelesen wird. Diese Aussagen lassen sich somit auf den diskreten Fall übertragen. Es ist festzuhalten, dass das Problem im diskreten Fall \mathcal{NP} -schwer im strengen Sinne ist, wenn die Drohne langsamer als der LKW ist und entweder nur Deadlines oder nur Release Dates vorliegen.

Ist hingegen die Drohne schneller als der LKW, so wird in diesem Abschnitt gezeigt, dass das Problem \mathcal{NP} -schwer im strengen Sinne ist, selbst wenn keine Zeitfenster vorgegeben sind und sich der Startort links oder rechts von allen übrigen Kunden befindet.

Lemma 4.9.1. *Es gelten die folgenden Gleichungen:*

$$\left(\frac{v^D + v^T}{v^D - v^T} \right) = 1 + \frac{2v^T}{v^D - v^T} \quad (4.37)$$

$$\left(\frac{v^D + v^T}{v^D - v^T} \right)^2 = 1 + \frac{4v^D v^T}{(v^D - v^T)^2} \quad (4.38)$$

$$\left(\frac{v^D + v^T}{v^D - v^T}\right)^3 = 1 + \frac{6v^{D^2}v^T + 2v^{T^3}}{(v^D - v^T)^3} \quad (4.39)$$

Beweis. Es gelten

$$\begin{aligned} \left(\frac{v^D + v^T}{v^D - v^T}\right) &= \frac{v^D - v^T - v^D + v^T + v^D + v^T}{v^D - v^T} = 1 + \frac{2v^T}{v^D - v^T} \\ \left(\frac{v^D + v^T}{v^D - v^T}\right)^2 &= \frac{(v^D - v^T)^2 - (v^D - v^T)^2 + (v^D + v^T)^2}{(v^D - v^T)^2} \\ &= 1 + \frac{-v^{D^2} + 2v^Dv^T - v^{T^2} + v^{D^2} + 2v^Dv^T + v^{T^2}}{(v^D - v^T)^2} \\ &= 1 + \frac{4v^Dv^T}{(v^D - v^T)^2} \\ \left(\frac{v^D + v^T}{v^D - v^T}\right)^3 &= \frac{(v^D + v^T)^3 + (v^D - v^T)^3 - (v^D - v^T)^3}{(v^D - v^T)^3} \\ &= 1 + \frac{(v^{D^2} + 2v^Dv^T + v^{T^2})(v^D + v^T)}{(v^D - v^T)^3} \\ &\quad - \frac{(v^{D^2} - 2v^Dv^T + v^{T^2})(v^D - v^T)}{(v^D - v^T)^3} \\ &= 1 + \frac{(v^{D^3} + 2v^{D^2}v^T + v^{T^2}v^D - v^{D^2}v^T + 2v^Dv^{T^2} + v^{T^3})}{(v^D - v^T)^3} \\ &\quad - \frac{(v^{D^3} - 2v^{D^2}v^T + v^{T^2}v^D - v^{D^2}v^T + 2v^Dv^{T^2} - v^{T^3})}{(v^D - v^T)^3} \\ &= 1 + \frac{6v^{D^2}v^T + 2v^{T^3}}{(v^D - v^T)^3}. \quad \square \end{aligned}$$

Lemma 4.9.2. Seien $v^D = 1, v^T = 1/(12m^4B^4)$ gegeben mit $B, m, k \in \mathbb{Z}_+$. Die folgenden Gleichungen gelten:

$$(v^D - v^T) = \frac{12m^4B^4 - 1}{12m^4B^4} \quad (4.40)$$

$$(v^D - v^T)^2 = \frac{144m^8B^8 - 24m^4B^4 + 1}{144m^8B^8} \quad (4.41)$$

$$(v^D - v^T)^3 = \frac{1728m^{12}B^{12} - 432m^8B^8 + 36m^4B^4 - 1}{1728m^{12}B^{12}} \quad (4.42)$$

Beweis.

$$\begin{aligned} (v^D - v^T) &= 1 - \frac{1}{12m^4B^4} = \frac{12m^4B^4 - 1}{12m^4B^4} \\ (v^D - v^T)^2 &= \frac{(12m^4B^4 - 1)(12m^4B^4 - 1)}{144m^8B^8} = \frac{144m^8B^8 - 24m^4B^4 + 1}{144m^8B^8} \\ (v^D - v^T)^3 &= \frac{(144m^8B^8 - 24m^4B^4 + 1)(12m^4B^4 - 1)}{1728m^{12}B^{12}} \\ &= \frac{1728m^{12}B^{12} - 432m^8B^8 + 36m^4B^4 - 1}{1728m^{12}B^{12}} \quad \square \end{aligned}$$

Lemma 4.9.3. Seien $v^D = 1, v^T = 1/(12m^4B^4)$ gegeben mit $B, m, k \in \mathbb{Z}_+, k \leq m$ sowie

$$\delta := \frac{6v^{D^2}v^T + 2v^{T^3}}{(v^D - v^T)^3}(kB + k) + \frac{B}{2}\left(\frac{4v^Dv^T}{(v^D - v^T)^2}\right) + \frac{B}{2}\left(\frac{2v^T}{v^D - v^T}\right).$$

Dann ist $\delta \in (0, 1)$.

Beweis. Folgende Aussagen werden gezeigt:

$$\frac{(6v^T + 2v^{T^3})(kB + k)}{(v^D - v^T)^3} < \frac{7}{10} \quad (4.43)$$

$$\frac{2Bv^T}{(v^D - v^T)^2} < \frac{1}{5} \quad (4.44)$$

$$\frac{Bv^T}{v^D - v^T} < \frac{1}{11}. \quad (4.45)$$

Sofern diese Abschätzungen gültig sind, lässt sich auch $\delta < 1$ ableiten. Zuerst wird (4.43) gezeigt. Es gilt

$$\begin{aligned} (8640m^8B^8 + 20)(kB + k) &< (8640m^8B^8 + 20)(mB + m) \\ &< (8640m^8B^8 + 20)(m^2B^2) < 8660m^{12}B^{12} \\ &= 12096m^{12}B^{12} - 3436m^{12}B^{12} \\ &< 12096m^{12}B^{12} - 3024m^8B^8 + 252m^4B^4 - 7 \end{aligned} \quad (4.46)$$

Da die Gleichungen $8640 = 10 \cdot 6 \cdot 144$, $12096 = 7 \cdot 1728$, $3024 = 7 \cdot 432$ und $252 = 7 \cdot 36$ gelten, ist (4.46) äquivalent zu

$$10(6(144m^8B^8) + 2)(kB + k) < 7(1728m^{12}B^{12} - 432m^8B^8 + 36m^4B^4 - 1), \quad (4.47)$$

bzw. zu

$$\frac{(6(144m^8B^8) + 2)(kB + k)}{(1728m^{12}B^{12} - 432m^8B^8 + 36m^4B^4 - 1)} < \frac{7}{10}. \quad (4.48)$$

Ferner gilt wegen (4.42) die folgende Gleichung:

$$\begin{aligned} \frac{(6v^T + 2v^{T^3})(kB + k)}{(v^D - v^T)^3} &= \frac{\left(6\frac{1}{v^{T^2}} + 2\right)(kB + k)v^{T^3}}{(1728m^{12}B^{12} - 432m^8B^8 + 36m^4B^4 - 1)v^{T^3}} \\ &= \frac{(6(144m^8B^8) + 2)(kB + k)}{1728m^{12}B^{12} - 432m^8B^8 + 36m^4B^4 - 1}. \end{aligned} \quad (4.49)$$

Mit (4.48) und (4.49) folgt nun (4.43).

Nun wird gezeigt, dass die Ungleichung (4.44) erfüllt ist.

$$\begin{aligned} 120m^4B^5 &= 120m^4B^5 + 24m^4B^5 - 24m^4B^5 < 144m^4B^5 - 24m^4B^4 + 1 \\ &< 144m^8B^8 - 24m^4B^4 + 1 \end{aligned}$$

Dies ist äquivalent zu

$$\begin{aligned} & 5(24m^4B^5) < 144m^8B^8 - 24m^4B^4 + 1 \\ \Leftrightarrow & \frac{(12m^4B^4)2B}{144m^8B^8 - 24m^4B^4 + 1} < \frac{1}{5} \Leftrightarrow \frac{\frac{2B}{v^T}}{144m^8B^8 - 24m^4B^4 + 1} < \frac{1}{5} \\ \Leftrightarrow & \frac{2Bv^T}{(144m^8B^8 - 24m^4B^4 + 1)v^{T^2}} \stackrel{(4.41)}{=} \frac{2B}{(v^D - v^T)^2} < \frac{1}{5} \end{aligned}$$

Also gilt (4.44). Abschließend wird (4.45) gezeigt:

$$\begin{aligned} 11B = 12B - B < 12B - 1 < 12m^4B^4 - 1 & \Leftrightarrow \frac{B}{12m^4B^4 - 1} < \frac{1}{11} \\ \Leftrightarrow \frac{\left(\frac{B}{12m^4B^4}\right)}{\left(\frac{12m^4B^4 - 1}{12m^4B^4}\right)} & \stackrel{(4.40)}{=} \frac{Bv^T}{(v^D - v^T)} < \frac{1}{11} \end{aligned}$$

Somit ist

$$\begin{aligned} \delta &= \frac{6v^{D^2}v^T + 2v^{T^3}}{(v^D - v^T)^3}(kB + k) + \frac{B}{2}\left(\frac{4v^Dv^T}{(v^D - v^T)^2}\right) + \frac{Bv^T}{2}\left(\frac{2v^T}{v^D - v^T}\right) \\ &< \frac{7}{10} + \frac{1}{5} + \frac{1}{11} = \frac{109}{110} < 1. \quad \square \end{aligned}$$

Satz 4.9.4. *Das Line-TSPwD mit diskreten Begegnungs- und Trennungsorten ohne Zeitfenster mit dem Ziel der Minimierung von $\max\{t_i^m, t_i\}$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Sei I eine Instanz des 3-Partition Problems, mit ganzen positiven Zahlen m und B und ganzen positiven Zahlen h_i für $i \in \{1, \dots, 3m\}$ mit $B/4 < h_i < B/2$ und $\sum_{i \in \{1, \dots, 3m\}} h_i = mB$. Mit $\Psi(I)$ wird die folgende Line-TSPwD Instanz mit diskreten Begegnungs- und Trennungsorten bezeichnet:

$$\begin{aligned} v^D &= 1 \\ v^T &= \frac{1}{12m^4B^4} \\ p_{i^*} &= (mB + m)v^T \\ p_i &= \frac{h_i(v^D - v^T) + 2v^T(mB + m)}{2} \quad \forall i \in \{1, \dots, 3m\} \\ p_i^{(k)} &= -v^T \left(\left(\frac{v^D + v^T}{v^D - v^T} \right) (kB + k) + 2 \left(\frac{p_i - v^T(mB + m)}{v^D - v^T} \right) \right) \\ &\quad + v^T(mB + m) \quad \forall i \in \{1, \dots, 3m\}, k \in \{0, \dots, m-1\} \\ p_i^{(k,j)} &= -v^T \left(\left(\frac{v^D + v^T}{v^D - v^T} \right)^2 (kB + k) + 2 \left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{p_j - v^T(mB + m)}{v^D - v^T} \right) \right) \\ &\quad + \frac{p_i - v^T(mB + m)}{(v^D - v^T)} \Bigg) + v^T(mB + m) \\ &\quad k \in \{0, \dots, m-1\}, \forall j \in \{1, \dots, 3m\}, \forall i \in H_j \\ p_k^{en} &= (mB + m)v^T - v^T(kB + k) \quad \forall k \in \{1, \dots, m\} \end{aligned}$$

Definiert wird folgende Menge $H_j := \{i \in \{1, \dots, 3m\} \mid h_i + h_j < \frac{3}{4}B\}$. Die Anzahl der Kunden ist nicht größer als $3m + m(3m + 9m^2) + m$. Es wird vorausgesetzt, dass der LKW den Kunden p_m^{en} bedient, also $T = \{p_m^{en}\}$. Die Entscheidungsfrage lautet „Gibt es eine zulässige Lösung, deren Zielfunktionswert nicht größer als $mB + m$?“. Eine Instanz wird als *zulässig* bezeichnet, wenn eine Lösung existiert, die mit *Ja* beantwortet wird. In einer Lösung einer solchen Instanz bezeichnet eine *Drohnen-Operation* ein Zeitintervall $[t_{\sigma(i)}^s, t_{\sigma(i)}^m]$. Es wird zunächst gezeigt, dass für eine zulässige Lösung für $\Psi(I)$ die folgenden beiden Aussagen wahr sind:

1. Der LKW fährt strikt von p_{i^*} nach p_m^{en} . Die Menge der möglichen Trennungs- und Begegnungsorte ist

$$\begin{aligned} & \{p_j^{en} \mid j \in \{1, \dots, m\}\} \cup \{p_i^{(k)} \mid k \in \{0, \dots, m-1\}, i \in \{1, \dots, 3m\}\} \\ & \cup \{p_i^{(k,j)} \mid k \in \{1, \dots, m-1\}, j \in \{1, \dots, 3m\}, i \in H_j\}. \end{aligned}$$

2. Die Drohne bedient die Kunden p_i für alle $i \in \{1, \dots, 3m\}$.

Wird die LKW-Tour der ersten Behauptung vorausgesetzt, so gelten Behauptung 3 bis 7.

3. Wenn die Drohne bei p_k^{en} gestartet wird, kann sie zwei Kunden nacheinander bedienen und sofort wieder gestartet werden, ohne dass Wartezeiten entstehen.
4. Werden drei Drohnen-Operationen mit den Kunden i, j, l im Zeitintervall $[kB + k, ((k+1)B + (k+1))]$ durchgeführt, so gilt

$$h_i + h_j + h_l \leq B.$$

5. In einem Zeitintervall $[kB + k, (k+1)B + (k+1)]$ sind maximal 3 Drohnen-Operationen möglich.
6. Werden in einem Zeitintervall $[kB + k, (k+1)B + (k+1)]$ genau 3 Drohnen-Operationen durchgeführt, so kann die Drohne erst ab dem Ort p_{k+1}^{en} erneut gestartet werden.
7. Wird die Drohne nicht zum Zeitpunkt p_k^{en} gestartet, so sind in den Zeitintervallen

$$[(k-1)B + (k-1), kB + k] \text{ und } [kB + k, (k+1)B + (k+1)]$$

jeweils maximal 2 Drohnen-Operationen möglich.

Da p_m^{en} durch den LKW bedient werden muss, fährt der LKW strikt in Richtung dieses Kunden, da

$$\frac{p_{i^*} - p_m^{en}}{v^T} = \frac{(mB + m)v^T - 0}{v^T} = mB + m$$

gilt. Also werden die Kunden $p_i^{(k)}$, $p_i^{(k,j)}$ und p_k^{en} ebenfalls durch den LKW bedient, wodurch sich die möglichen Begegnungs- und Trennungsorte ableiten

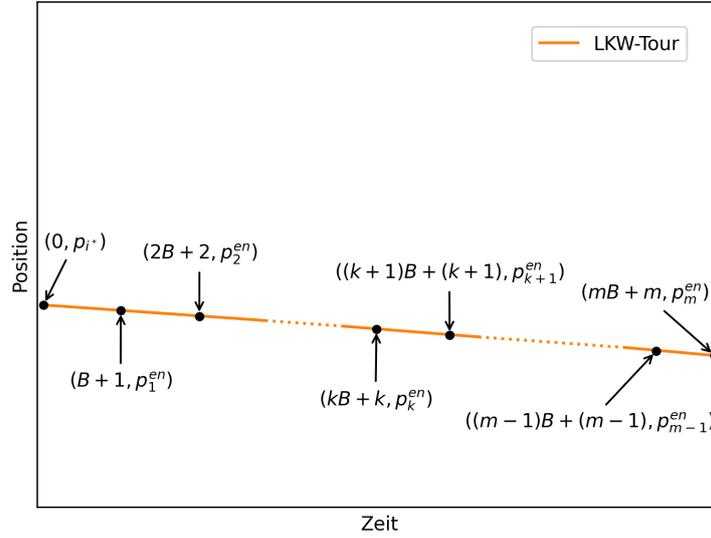


Abbildung 4.16: Erzwungene LKW-Tour

lassen. Somit ist die erste Behauptung bewiesen.

Die LKW-Tour wird in Abbildung 4.16 dargestellt. Da $p_i > v^T(mB + m)$ ist, werden die Kunden p_i für alle $i \in \{1, \dots, 3m\}$ durch die Drohne bedient, um eine *Ja*-Lösung zu ermöglichen. Die zweite Behauptung ist somit korrekt.

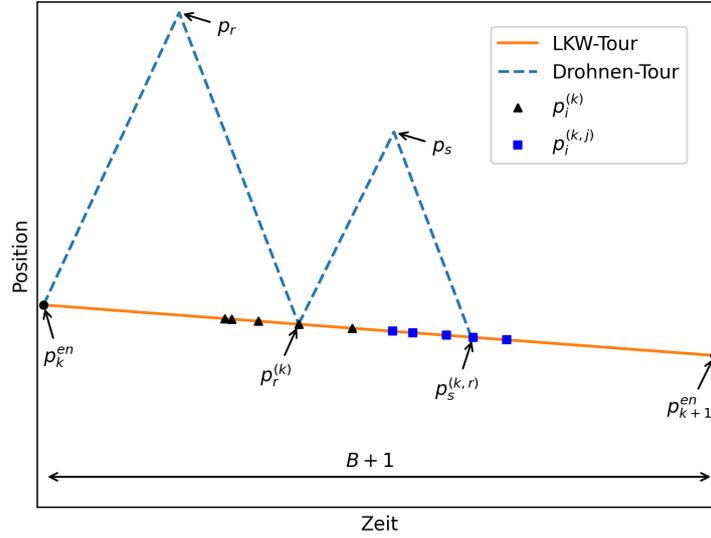
Wird die Drohne am Ort p_k^{en} gestartet, um den Kunden p_i zu bedienen, so begegnen sich Drohne und LKW am Ort $p_i^{(k)}$. Für die Drohne entstehen hierbei keine Wartezeiten. Dies lässt sich mit Lemma 4.4.15 für $t_i^s = kB + k$ zeigen:

$$\begin{aligned}
 t_i^m &= \left(\frac{v^D + v^T}{v^D - v^T} \right) (kB + k) + 2 \left(\frac{p_i - (mB + m)v^T}{v^D - v^T} \right) \\
 &= \left(\frac{v^D + v^T}{v^D - v^T} \right) (kB + k) \\
 &\quad + 2 \left(\frac{\left(\frac{1}{2} (h_i(v^D - v^T) + 2v^T(mB + m)) \right) - (mB + m)v^T}{v^D - v^T} \right) \\
 &= \left(\frac{v^D + v^T}{v^D - v^T} \right) (kB + k) + h_i
 \end{aligned}$$

Zum Zeitpunkt t_i^m befindet sich der LKW am Ort

$$f^T(t_i^m) = -v^T \left(\left(\frac{v^D + v^T}{v^D - v^T} \right) (kB + k) + h_i \right) + v^T(mB + m) = p_i^{(k)}.$$

Wird die Drohne unmittelbar wieder gestartet, um den Kunden j zu bedienen und anschließend zum LKW zurückzukehren, trifft die Drohne den LKW an der Position $p_j^{(k,i)}$, an der sie auch unmittelbar wieder gestartet werden könnte.

Abbildung 4.17: Zwei Drohnen-Operationen ab p_k^{en}

Auch hier findet Lemma 4.4.15 Anwendung für $t_i^s = kB + k$ sowie dem Drohnen-Schedule $\sigma = (i, j)$.

$$\begin{aligned} t_j^m &= \left(\frac{v^D + v^T}{v^D - v^T} \right)^2 (kB + k) \\ &\quad + 2 \left(\left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{p_j - (mB + m)v^T}{v^D - v^T} \right) + \frac{p_i - (mB + m)v^T}{v^D - v^T} \right) \\ &= \left(\frac{v^D + v^T}{v^D - v^T} \right)^2 (kB + k) + \left(\frac{v^D + v^T}{v^D - v^T} \right) h_j + h_i \end{aligned}$$

Zu diesem Zeitpunkt befindet sich der LKW am Ort:

$$\begin{aligned} f^T(t_i^m) &= -v^T \left(\left(\frac{v^D + v^T}{v^D - v^T} \right)^2 (kB + k) + \left(\frac{v^D + v^T}{v^D - v^T} \right) h_j + h_i \right) \\ &\quad + v^T(mB + m) = p_j^{(k,i)}. \end{aligned}$$

Diese Überlegungen werden in Abbildung 4.17 veranschaulicht. Somit ist die dritte Behauptung korrekt.

Im nächsten Schritt wird gezeigt, dass die Ungleichung $h_{\sigma(1)} + h_{\sigma(2)} + h_{\sigma(3)} \leq B$ erfüllt sein muss, falls sich die Drohne ab dem Zeitpunkt $kB + k$ am LKW befindet und drei Drohnenkunden $\sigma(1), \sigma(2), \sigma(3)$ bedient werden sollen, sodass zum Zeitpunkt $B(k + 1) + (k + 1)$ die Drohne wieder beim LKW ist. Wegen 4.4.15 gilt:

$$t_{\sigma(3)}^m = (kB + k) \left(\frac{v^D + v^T}{v^D - v^T} \right)^3 + 2 \left(\left(\frac{v^D + v^T}{v^D - v^T} \right)^2 \left(\frac{p_{\sigma(1)} - v^T(mB + m)}{v^D - v^T} \right) \right)$$

$$\begin{aligned}
& + \left(\frac{v^D + v^T}{v^D - v^T} \right) \left(\frac{p_{\sigma(2)} - v^T(mB + m)}{v^D - v^T} \right) + \left(\frac{p_{\sigma(3)} - v^T(mB + m)}{v^D - v^T} \right) \Bigg) \\
& = (kB + k) \left(\frac{v^D + v^T}{v^D - v^T} \right)^3 + \left(\frac{v^D + v^T}{v^D - v^T} \right)^2 h_{\sigma(1)} + \left(\frac{v^D + v^T}{v^D - v^T} \right) h_{\sigma(2)} \\
& \quad + h_{\sigma(3)} \\
& \stackrel{4.9.1}{=} \underbrace{\frac{6v^{D^2}v^T + 2v^{T^3}}{(v^D - v^T)^3} (kB + k) + \frac{4v^D v^T}{(v^D - v^T)^2} h_{\sigma(1)} + \frac{2v^T}{(v^D - v^T)} h_{\sigma(2)}}_{=: \tilde{\delta}} \quad (4.50) \\
& \quad + (kB + k + h_{\sigma(1)} + h_{\sigma(2)} + h_{\sigma(3)})
\end{aligned}$$

Es handelt sich also um eine untere Schranke für den Begegnungszeitpunkt von Drohne und LKW, nachdem die Drohne die drei Drohnenkunden bedient hat. Es soll nun gelten

$$\begin{aligned}
\tilde{\delta} + (kB + k + h_{\sigma(1)} + h_{\sigma(2)} + h_{\sigma(3)}) &\leq (k + 1)B + (k + 1) \\
\Leftrightarrow \tilde{\delta} + (h_{\sigma(1)} + h_{\sigma(2)} + h_{\sigma(3)}) &\leq B + 1 \\
\Leftrightarrow 1 + (h_{\sigma(1)} + h_{\sigma(2)} + h_{\sigma(3)}) &\leq B + 1 \\
\Leftrightarrow h_{\sigma(1)} + h_{\sigma(2)} + h_{\sigma(3)} &\leq B.
\end{aligned}$$

Diese Umformungen gelten, da $h_{\sigma(1)}, h_{\sigma(2)}, h_{\sigma(3)}$ und B ganze Zahlen sind. Wegen 4.9.3 und $h_{\sigma(1)}, h_{\sigma(2)} < B/2$ ist $0 < \tilde{\delta} < 1$. Damit gilt Behauptung 4. Die 5. Behauptung folgt mit

$$\begin{aligned}
t_{\sigma(4)}^m &= (kB + k) \left(\frac{v^D + v^T}{v^D - v^T} \right)^4 + \left(\frac{v^D + v^T}{v^D - v^T} \right)^3 h_{\sigma(1)} + \left(\frac{v^D + v^T}{v^D - v^T} \right)^2 h_{\sigma(2)} \\
& \quad + \left(\frac{v^D + v^T}{v^D - v^T} \right) h_{\sigma(3)} + h_{\sigma(4)} > (kB + k) + \underbrace{h_{\sigma(1)} + h_{\sigma(2)} + h_{\sigma(3)} + h_{\sigma(4)}}_{\geq B+1} \\
& \geq kB + k + B + 1 = (k + 1)B + k + 1.
\end{aligned}$$

Nun wird Behauptung 6 gezeigt. Es gilt

$$\begin{aligned}
t_{\sigma(3)}^m &\stackrel{4.50}{=} \tilde{\delta} + kB + k + h_{\sigma(1)} + h_{\sigma(2)} + h_{\sigma(3)} \stackrel{h_i > B/4}{>} \tilde{\delta} + kB + k + \frac{3}{4}B \\
&\stackrel{4.50}{=} \frac{6v^{D^2}v^T + 2v^{T^3}}{(v^D - v^T)^3} (kB + k) + \frac{4v^D v^T}{(v^D - v^T)^2} h_{\sigma(1)} + \frac{2v^T}{(v^D - v^T)} h_{\sigma(2)} \\
& \quad + kB + k + B \frac{3}{4} \\
&> \frac{4v^D v^T}{(v^D - v^T)^2} (kB + k) + \frac{4v^D v^T}{(v^D - v^T)^2} \frac{B}{4} + \frac{2v^T}{(v^D - v^T)} \frac{B}{4} + kB + k + B \frac{3}{4} \\
&> \frac{4v^D v^T}{(v^D - v^T)^2} (kB + k) + \frac{2v^T}{(v^D - v^T)} \frac{B}{2} + kB + k + B \frac{3}{4} \\
&\stackrel{4.9.1}{=} \left(\frac{v^D + v^T}{v^D - v^T} \right)^2 (kB + k) + \left(\frac{v^D + v^T}{v^D - v^T} \right) \frac{B}{2} + \frac{B}{4}
\end{aligned}$$

$$> \max_{\{i,j \in \{1, \dots, 3m\} : h_i + h_j < \frac{3}{4}B\}} (kB + k) \left(\frac{v^D + v^T}{v^D - v^T} \right)^2 + \left(\left(\frac{v^D + v^T}{v^D - v^T} \right) h_j + h_i \right).$$

Insgesamt folgt

$$t_{\sigma(3)}^m > \max_{\{i,j \in \{1, \dots, 3m\} : h_i + h_j < \frac{3}{4}B\}} (kB + k) \left(\frac{v^D + v^T}{v^D - v^T} \right)^2 + \left(\left(\frac{v^D + v^T}{v^D - v^T} \right) h_j + h_i \right).$$

Dies bedeutet, dass innerhalb eines $kB + k$ Intervalls, in dem drei Drohnenkunden bedient wurden, die Drohne beim Kunden p_k^{en} warten muss, um wieder einen nächsten Kunden bedienen zu können. Also ist auch die 6. Behauptung richtig.

Im Falle einer Drohnen-Operation mit Startzeitpunkt x und Endzeitpunkt y , wobei $x < kB + k < y$ gilt, können wegen der sechsten Behauptung in Zeitintervall $[(k-1)B + (k-1), kB + k]$ keine drei Drohnen-Operationen stattfinden. Im Zeitintervall $[kB + k, (k+1)B + (k+1)]$ kann die Drohne frühestens ab dem Ort

$$z = \min_{i \in \{1, \dots, 3m\}} p_i^{(k)}$$

erneut gestartet werden. Die Zeitspanne, die zwischen dem Besuch des LKWs bei p_k^{en} und z liegt, entspricht einer Drohnen-Operation. Es wurde nachgewiesen, dass maximal drei Drohnen-Operationen im Zeitintervall $[kB + k, (k+1)B + (k+1)]$ möglich sind (6. Behauptung). Also sind nur noch zwei Drohnen-Operationen möglich, falls die Drohne nicht bei p_k^{en} gestartet wurde. Somit folgt auch die 7. Behauptung.

Sei eine zulässige Lösung einer 3-Partition-Instanz I gegeben mit den Mengen A_1, \dots, A_m und $\varphi^k(1), \varphi^k(2), \varphi^k(3) \in A_k$. Wegen der 4. Behauptung ist es der Drohne möglich, innerhalb des Zeitintervalls $[kB + k, (k+1)B + (k+1)]$ die Kunden $p_{\varphi^k(1)}, p_{\varphi^k(2)}, p_{\varphi^k(3)}$ zu bedienen.

Es sei nun eine zulässige Lösung einer Line-TSPwD-Instanz $\Psi(I)$ gegeben. Dann existieren m Zeitintervalle $[kB + k, (k+1)B + (k+1)]$, wobei $k \in \{0, \dots, m-1\}$ gilt. Die Drohne bedient alle Kunden p_i mit $i \in \{1, \dots, 3m\}$. Sei

$$D := \{p_k^{en} \mid k \in \{1, \dots, m\} \text{ und Drohne trifft an diesem Ort nicht auf den LKW}\}.$$

Unter der Voraussetzung, dass $|D| \geq 1$ ist, folgt aus der 7. Behauptung, dass mindestens $|D| + 1$ Intervalle existieren, in denen maximal zwei Drohnen-Operationen durchgeführt werden können. In den übrigen Intervallen werden maximal 3 Drohnen-Operationen durchgeführt (Behauptung 5). Insgesamt gibt es m Intervalle. Daher ist die maximale Anzahl an Drohnen-Operationen

$$3(m - (|D| + 1)) + |D| + 2(|D| + 1) = 3m - 3|D| - 3 + |D| + 2|D| + 2 = 3m - 1.$$

Jedoch sind $3m$ viele Kunden durch die Drohne zu bedienen. Somit müssen sich Drohne und LKW für alle $k \in \{1, \dots, m\}$ bei p_k^{en} begegnen. Innerhalb jedes dieser Zeitintervalle müssen demnach drei Drohnen-Operationen durchgeführt werden. Um dies zu gewährleisten, muss für jedes Zeitintervall $[kB + k, (k+1)B + (k+1)]$, in dem die Kunden $p_{\sigma^k(1)}, p_{\sigma^k(2)}, p_{\sigma^k(3)}$ durch die Drohne bedient werden, die Ungleichung

$$h_{\sigma^k(1)} + h_{\sigma^k(2)} + h_{\sigma^k(3)} \leq B$$

erfüllt sein (4. Behauptung). Insgesamt folgt nun die Aussage

$$\forall k \in \{1, \dots, m\} : h_{\sigma^k(1)} + h_{\sigma^k(2)} + h_{\sigma^k(3)} = B$$

wegen $\sum h_i = mB$ und der Tatsache, dass es genau m Zeitintervalle gibt. Somit ist auch I zulässig. \square

Satz 4.9.5. *Das Line-TSPwD mit diskreten Begegnungs- und Trennungsorten ohne Zeitfenster mit dem Ziel der Minimierung von $\max\{t_i\}$ ist \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Es wird auf den Beweis von 4.9.4 verwiesen. Zusätzlich werden noch zwei Kunden

$$p_{last}^D = mB + m \text{ und } p_{last}^T = 2p_{last}^D \left(\frac{v^T}{v^T - 1} \right)$$

hinzugefügt, wobei p_{last}^T nur durch den LKW bedient werden kann. Ferner sei der Schwellwert in $\Psi(I)$ gleich $mB + m + \frac{|p_{mB+m}^{en} - p_{last}^T|}{v^T}$. Dann fährt der LKW strikt von p_{i^*} zu p_{last}^T , da $p_{last}^T < 0 = p_{mB+m}^{en}$ gilt. Wird die Drohne zum Zeitpunkt 0 sofort gestartet, um den Kunden p_{last}^D zu bedienen, so könnte sie erst nach p_{mB+m}^{en} aufgelesen werden.:

$$\begin{aligned} \frac{p_{last}^D - (mB + m)v^T}{v^D} + \frac{p_{last}^D - p_{mB+m}^{en}}{v^D} \\ = mB + m - (mB + m)v^T + mB + m - 0 \\ = (2 - v^T)(mB + m) > mB + m. \end{aligned}$$

Danach gibt es nur noch einen Kundenort, an dem die Drohne gestartet oder abgeholt werden kann. Dies kann nicht zu einer *Ja*-Antwort führen. Daher muss p_{last}^D als letzter Kunde durch die Drohne bedient werden. Wird sie bei p_{mB+m}^{en} gestartet, so erreicht sie zum Schwellwert Kunde p_{last}^D . \square

Korollar 4.9.6. *Das Drohnen Scheduling Problem für eine gegebene LKW-Route (DSP) mit einer Drohne (Boysen et al., 2018) ist für die Minkowski-Distanz mit $1 \leq p \leq \infty$ \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Analog zu 4.9.4. \square

(Tang et al., 2019) bezieht sich auf das TSP-D in (Agatz et al., 2018), wobei dort die Kunden Knoten auf einem Graphen sind und die Kantengewichte die Dreiecksungleichung erfüllen. In (Agatz et al., 2018) ist es erlaubt, einen LKW-Kunden mehrmals zu besuchen, auch wenn dieser schon bedient wurde, um dort die Drohne zu starten. Dies wird auch in der vorliegenden Arbeit so gehandhabt. In (Tang et al., 2019) ist dies jedoch ausgeschlossen. Der CP-Ansatz erlaubt ein Auflesen oder Starten nur, wenn der LKW-Kunde zu diesem Zeitpunkt auch bedient wird. Dies ist analog zum FSTSP (Murray & Chu, 2015). Darüber hinaus wird in (Tang et al., 2019) eine Metrik zwischen den Knoten vorausgesetzt. Dies bedeutet, dass ein Ort (Kunde/Knoten) nur einmal bedient werden kann. An einem Ort kann es nicht zwei verschiedene Kunden geben, da für eine Metrik d auf einer Menge X die Äquivalenz $d(x, y) = 0 \Leftrightarrow x = y$ gilt.

Korollar 4.9.7. *Das Line-TSPwD mit diskreten Begegnungs- und Trennungsorten, wobei der LKW wieder zum Startort zurückkehren muss, sowie das Drone Routing Subproblem (DRS) (Tang et al., 2019), das FSTSP (Murray & Chu, 2015) und das TSP-D (Agatz et al., 2018), wobei sich die Kunden auf einer Linie befinden, sind für die Minkowski-Distanz mit $1 \leq p \leq \infty$ \mathcal{NP} -schwer im strengen Sinne.*

Beweis. Der Beweis erfolgt grundsätzlich analog zum Beweis von Satz 4.9.4. Allerdings ist zu berücksichtigen, dass der LKW bei einem Depot startet und auch dort endet. Darüber hinaus ist die LKW-Tour beim DRS fest vorgegeben sowie im Line-TSPwD erzwungen, weshalb einige Modifikationen erforderlich sind. Es werden zusätzlich zu $\Psi(I)$ im Beweis von 4.9.4 vier Kunden hinzugefügt, von denen zwei durch die Drohne bedient werden. Die feste bzw. erzwungene LKW-Tour ist

$$f^T(t) = \begin{cases} (mB + m)v^T - v^T \cdot t & \text{if } 0 \leq t \leq 3(mB + m) \\ v^T \cdot t - 5(mB + m)v^T & \text{if } t > 3(mB + m). \end{cases} \quad (4.51)$$

Die Orte der vier zusätzlichen Kunden sind

$$\begin{aligned} p_1^{DRS} &= -2(mB + m) + \frac{1 - v^T}{2} & p_2^{DRS} &= \frac{(mB + m)(v^D + v^T) + v^D - v^T}{2} \\ p_3^{DRS} &= -2(mB + m)v^T & p_4^{DRS} &= -v^T, \end{aligned}$$

wobei p_3^{DRS} und p_4^{DRS} durch den LKW bedient werden müssen. Die Entscheidungsfrage lautet „Gibt es eine zulässige Lösung, deren Zielfunktionswert $6(mB + m)$ nicht übersteigt?“. Wenn die Rückkehr der Drohne und des LKWs zum Startkunden gefordert wird, nachdem alle Kunden bedient wurden, wird die in (4.51) angegebene Tour erzwungen, da

$$2 \left(\frac{p_{i^*} - p_3^{DRS}}{v^T} \right) = 2 \left(\frac{(mB + m)v^T + 2(mB + m)v^T}{v^T} \right) = 6(mB + m).$$

gilt. Im Folgenden werden alle Kunden aus der Reduktion des Beweises von Satz 4.9.4 übernommen. Für eine zulässige Lösung von $\Psi(I)$ gelten folgende Behauptungen

1. Die Drohne muss spätestens zum Zeitpunkt $mB + m$ starten, um den Kunden p_1^{DRS} zu bedienen, damit sie zum Zeitpunkt $5(mB + m) - 1$ erneut starten kann. Durch einen früheren Start kann die Drohne jedoch nicht früher erneut gestartet werden.
2. Die Drohne muss spätestens zum Zeitpunkt $mB + m + 1$ starten, um den Kunden p_1^{DRS} zu bedienen, damit sie zum Zeitpunkt $5(mB + m)$ erneut starten kann. Durch einen früheren Start kann die Drohne jedoch nicht früher erneut gestartet werden.
3. Die Drohne muss im Intervall $[0, mB + m + 1]$ starten, um den Kunden p_1^{DRS} zu bedienen.
4. Die Drohne wird zum Zeitpunkt 0 oder $5(mB + m) - 1$ gestartet, um den Kunden p_2^{DRS} zu bedienen. Wenn die Drohne zum Zeitpunkt 0 startet,

kann sie zum Zeitpunkt $mB + m + 1$ erneut starten. Findet der Start zum Zeitpunkt $5(mB + m) - 1$ statt, so kann die Drohne zum Zeitpunkt $6(mB + m)$ wieder aufgelesen werden.

Der Kunde p_1^{DRS} ist bereits bei einem Start zum Zeitpunkt 0 ein Wendekunde, d.h. es gilt $0 \in T_2(p_1^{DRS})$. Aus Lemma 4.4.4 folgt, dass der Begegnungszeitpunkt von Drohne und LKW nach dem Zeitpunkt $3(mB + m)$ liegt.

$$\begin{aligned}
& 2 \left(\frac{2(mB + m) - \frac{1-v^T}{2} + (mB + m)v^T}{v^T + v^D} \right) \\
&= \frac{4(mB + m) + 2v^T(mB + m)}{v^T + 1} - \frac{1 - v^T}{2(v^T + 1)} \\
&= \frac{4(mB + m) + \frac{2(mB+m)}{12m^4 \cdot B^4}}{\frac{12m^4 \cdot B^4 + 1}{12m^4 \cdot B^4}} - \frac{1 - v^T}{2(v^T + 1)} \\
&= \frac{12m^4 \cdot B^4 \cdot (4(mB + m)) + 2(mB + m)}{12m^4 \cdot B^4 + 1} - \frac{1 - v^T}{2(v^T + 1)} \\
&> \frac{12m^4 \cdot B^4}{12m^4 \cdot B^4 + 1} \cdot 4(mB + m) - \frac{1}{2} \geq \frac{12}{13} \cdot 4(mB + m) - \frac{1}{2} \\
&\geq \frac{96 - 13}{26} \cdot (mB + m) > 3(mB + m)
\end{aligned}$$

Da lediglich zwei Kunden zwischen $mB + m$ und $5(mB + m)$ besucht werden, nämlich der Kunde p_3^{DRS} zum Zeitpunkt $3(mB + m)$ und der Kunde p_4^{DRS} zu den Zeitpunkten $mB + m + 1$ und $5(mB + m) - 1$, kann die Drohne frühestens zum Zeitpunkt $5(mB + m) - 1$ vom LKW aufgelesen werden. Wird die Drohne zum Zeitpunkt $mB + m$ gestartet, um den Kunden p_1^{DRS} zu bedienen, so begegnen sich Drohne und LKW genau zum Zeitpunkt

$$\begin{aligned}
t_{p_1^{DRS}}^m &= \frac{2 \left(2(mB + m) - \frac{1-v^T}{2} - 2(mB + m)v^T \right)}{v^D - v^T} + mB + m \\
&= \frac{4(mB + m)(v^D - v^T) - (1 - v^T)}{v^D - v^T} + mB + m = 5(mB + m) - 1.
\end{aligned}$$

Analog begegnen sich Drohne und LKW zum Zeitpunkt $5(mB + m)$, falls die Drohne zum Zeitpunkt $mB + m + 1$ gestartet ist. Durch ein früheres Starten kann die Drohne nicht früher aufgelesen werden. Somit folgen Behauptung 1 und 2.

Wird die Drohne erst ab dem Zeitpunkt $5(mB + m) - 1$ gestartet, so wird sie wegen der Symmetrie nach $6(mB + m)$ abgeholt. Also gibt es keine Lösung, deren Zielfunktionswert $6(mB + m)$ nicht überschreitet. Also muss die Drohne im Intervall $[0, mB + m]$ gestartet werden, um den Kunden p_1^{DRS} zu bedienen. Damit gilt Behauptung 3.

Im Folgenden wird geprüft, zu welchen Zeiten die Drohne gestartet werden kann, um den Kunden p_2^{DRS} zu bedienen. Wird die Drohne vor $mB + m + 1$ gestartet, muss sie spätestens zum Zeitpunkt $mB + m + 1$ am LKW sein, da zu diesem Zeitpunkt die Drohne wieder für den Kunden p_1^{DRS} benötigt wird. Es wird sich zeigen, dass dies ausschließlich für den Zeitpunkt 0 möglich ist.

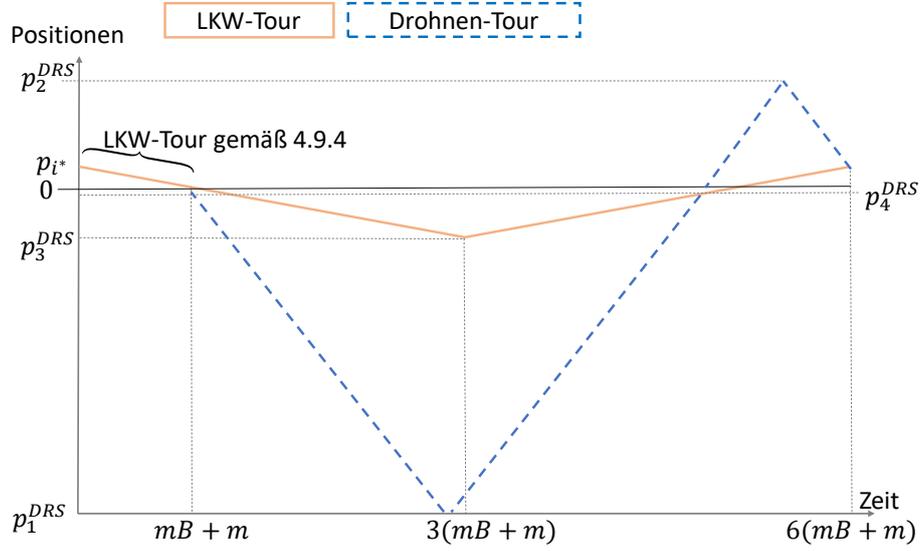


Abbildung 4.18: Lösung einer Instanz nach Reduktion

Wegen Lemma 4.4.16 ist der früheste Begegnungszeitpunkt von Drohne und LKW gleich

$$2 \left(\frac{\left(\frac{(mB+m)(v^D+v^T)+v^D-v^T}{2} \right) - (mB+m)v^T}{v^D-v^T} \right) =$$

$$\frac{(mB+m)(v^D+v^T)+v^D-v^T-2v^T(mB+m)}{(v^D-v^T)} = \frac{(mB+m)(v^D-v^T)}{(v^D-v^T)} + 1$$

$$= mB+m+1.$$

Für den Drohneneinsatz, bei dem der Kunde p_2^{DRS} bedient wird, bleiben nur die Zeitintervalle $[0, mB+m+1]$ oder $[5(mB+m)-1, 6(mB+m)]$. Somit folgt Behauptung 4. Die übrigen Kunden werden von der Drohne in den Zeitintervallen $[0, mB+m]$ oder $[5(mB+m), 6(mB+m)]$ bedient. Diese enthalten exakt die gleichen Begegnungs- und Trennorte wie im Beweis von Lemma 4.9.4. Für den weiteren Beweis kann auf die Reduktion in Satz 4.9.4 verwiesen werden. \square

Es stellt sich die Frage, ob möglicherweise die Transformationen aus (Boysen et al., 2018) oder (Tang et al., 2019) ausreichen, um die \mathcal{NP} -Schwere im strengen Sinne für die Minkowski-Norm für ein beliebiges p zu zeigen. Es wird nun gezeigt dass dies nicht unmittelbar möglich ist.

Satz 4.9.8. *Die pseudopolynomiellen Transformationen in (Tang et al., 2019) und (Boysen et al., 2018) lassen sich nicht auf \mathbb{R}^1 und \mathbb{R}^2 mit der Minkowski-Distanz übertragen.*

Beweis. Hier wird gezeigt, dass sich die gegebenen Graphen nicht in \mathbb{R}^n für $n \in \{1, 2\}$ mit der Minkowski-Distanz einbetten lassen. Die Frage, ob solch eine Einbettung existiert wird auch als *Distance Geometry Problem* bezeichnet (siehe (Lavor, Liberti, Lodwick & Mendonça da Costa, 2017)).

Zunächst wird gezeigt, dass solch eine Einbettung für (Boysen et al., 2018) nicht funktioniert. Die Fahrzeit des LKWs von einem Kunden zum nächsten ist 0. Das bedeutet, dass im \mathbb{R}^n die Menge der Orte, die der LKW besucht, einelementig wäre und der LKW sich nicht bewegt. Daher kann sich die Entfernung zwischen dem LKW und den Kunden, die von der Drohne besucht werden, nicht ändern. In jeder Reihenfolge der von der Drohne besuchten Kunden wird derselbe Zielfunktionswert erreicht.

In (Tang et al., 2019) wird eine 3-Partition Instanz durch eine pseudopolynomielle Transformation Ψ in eine DRS Instanz überführt. Sei eine 3-Partition Instanz I mit den Parametern $B, m, h_i \in \mathbb{N}$ mit $B/4 < h_i < B/2$ für alle $i \in \{1, \dots, 3m\}$ sowie $\sum_{i \in \{1, \dots, 3m\}} h_i = mB$ gegeben. Dann gibt es $mB + m$ LKW-Stopps in $\Psi(I)$. Dem LKW wird die Tour $k_i \rightarrow k_{i+1}$ mit Startort k_1 und Zielort $k_{mB+m} = k_1$ mit den Stopps k_i mit $i \in \{1, \dots, mB + m\}$ zugewiesen, wobei alle k_i verschieden sind, bis auf den Start- und Endort (Depot). Die Fahrzeit des LKW von $k_i \rightarrow k_{i+1}$ beträgt $1/2m$, wenn i von $B + 1$ geteilt wird und sonst ist die Fahrzeit 1. Außerdem muss die Drohne $4m$ Kunden c_1, \dots, c_{4m} bedienen, wobei die Reisezeit der Drohne von jedem Stopp des LKW zu c_i mit $i \leq 3m$ genau $h_i/2$ beträgt. Die Reisezeiten für den Hin- und Rückweg sind stets identisch. Die Reisezeit der Drohne nach c_i mit $i > 3m$ beträgt für die Haltestellen k_{i-3m+1} und k_{i-3m+2} jeweils genau $1/4m$ und für alle übrigen Haltestellen genau eine Zeiteinheit. Da alle vom LKW angefahrenen Orte den gleichen Abstand von c_i mit $i \leq 3m$ haben, müssen k_1, \dots, k_{mB+m} in der Schnittmenge der jeweiligen Kreise liegen.

In \mathbb{R}^1 ist für jedes p die Minkowski-Distanz gleich dem Absolutbetrag. In diesem Fall ist die Schnittmenge zweier verschiedener Kreise höchstens 1-elementig. Gegeben seien zwei Kreise mit unterschiedlichen positiven Radien $r_1 \neq r_2$ und Mittelpunkten c_1, c_2 . Mit

$$S^1 = \{x \in \mathbb{R} \mid |x - c_1| = r_1\} \text{ und } S^2 = \{x \in \mathbb{R} \mid |x - c_2| = r_2\}$$

ist $S^1 \cap S^2 = \{c_1 - r_1, c_1 + r_1\} \cap \{c_2 - r_2, c_2 + r_2\}$. Angenommen es wäre $|S^1 \cap S^2| = 2$, dann gilt

$$[(c_1 - r_1 = c_2 - r_2 \wedge c_1 + r_1 = c_2 + r_2) \vee (c_1 - r_1 = c_2 + r_2 \wedge c_1 + r_1 = c_2 - r_2)] \\ \wedge (r_1 \neq r_2) \wedge (r_1, r_2 \geq 0).$$

Dies ist äquivalent zu

$$[(r_1 = r_2 \wedge c_1 = c_2) \vee (r_1 = -r_2 \wedge c_1 = c_2)] \wedge (r_1 \neq r_2) \wedge (r_1, r_2 \geq 0). \quad (4.52)$$

Die Aussage (4.52) ist offensichtlich falsch. Somit enthält der Schnitt zweier unterschiedlicher Kreise im \mathbb{R}^1 mit der Minkowski-Distanz maximal ein Element. Also gibt es bei der LKW-Tour nur einen Stopp. Jedoch sind $B(m+1)$ verschiedene Stopps notwendig.

Im Falle der Minkowski-Distanz mit $2 \leq p < \infty$ in \mathbb{R}^2 kann die Schnittmenge maximal 2 Elemente enthalten ((Poodiack & Wood, 2022), Theorem 18.2). Dies

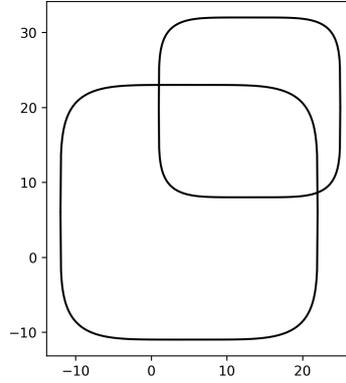


Abbildung 4.19: Zwei Kreise mit Mittelpunkten $m_1 = (13, 20)$, $m_2 = (5, 6)$ und Radien $r_1 = 12$, $r_2 = 17$ bezüglich der Minkowski-Distanz mit $p = 5$.

wird in Abbildung 4.19 beispielhaft illustriert. Für den Fall, dass $p \notin \{1, 2, \infty\}$, gilt sogar, dass für rationale Mittelpunkte und rationale Radien maximal nur ein rationaler Schnittpunkt existiert. Dies folgt unmittelbar aus dem Großen Satz von Fermat (Wiles, 1995). Benötigt werden aber $B(m + 1)$ verschiedene Orte.

Betrachtet wird nun wieder die Ebene \mathbb{R}^2 mit der Minkowski-Distanz für $p = 1$ (für $p = \infty$ gilt das Folgende analog). Im Gegensatz zu $1 < p < \infty$ kann die Schnittmenge mehrerer unterschiedlicher Kreise unendlich viele Elemente enthalten. Dies lässt sich an Abbildung 4.20 erkennen. Ein Kreis mit Mittelpunkt (a, b) und Radius r hat die folgende Struktur

$$\begin{aligned} S &= \{(x, y) \mid |x - a| + |y - b| = r\} \\ &= \{(x, y) \mid y = -x + a + b + r \wedge x \in [a, a + r]\} \\ &\quad \cup \{(x, y) \mid y = x - a + b + r \wedge x \in [a - r, a]\} \\ &\quad \cup \{(x, y) \mid y = x - a + b - r \wedge x \in [a, a + r]\} \\ &\quad \cup \{(x, y) \mid y = -x + a + b - r \wedge x \in [a - r, a]\}. \end{aligned}$$

Für zwei verschiedene Kreise S^1 und S^2 gilt $|S^1 \cap S^2| \in \{0, 1, 2, \infty\}$. Oben wurde gezeigt, dass mindestens 3 verschiedene Stopps für den LKW notwendig sind. Damit dies möglich ist, hat die Schnittmenge der Kreise die Struktur

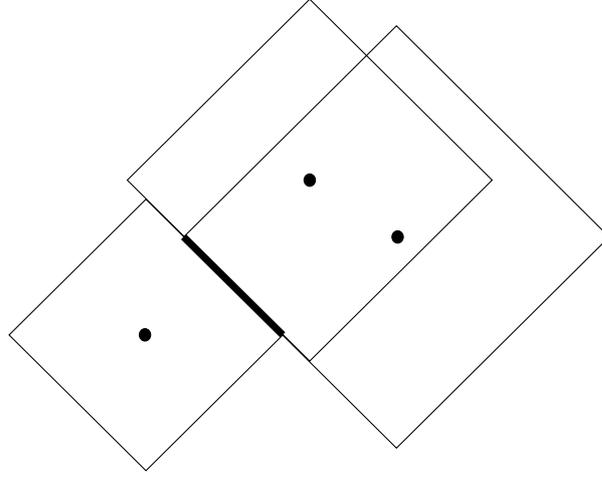
$$A := \{(x, y) \mid y = x + c \wedge x \in [e, f]\} \text{ oder } B := \{(x, y) \mid y = -x + c \wedge x \in [e, f]\}.$$

Für zwei Elemente $(a_1, b_1), (a_2, b_2)$ aus A ist die Distanz

$$d((a_1, b_1), (a_2, b_2)) = |a_1 - a_2| + |b_1 - b_2| = |a_1 - a_2| + |a_1 + c - a_2 - c| = 2|a_1 - a_2|.$$

Somit gilt für 3 Elemente $(a_1, b_1), (a_2, b_2), (a_3, b_3)$ aus A mit $a_1 \geq a_2 \geq a_3$, dass

$$d((a_1, b_1), (a_3, b_3)) = d((a_1, b_1), (a_2, b_2)) + d((a_2, b_2), (a_3, b_3))$$

Abbildung 4.20: Schnittmenge dreier Kreise in \mathbb{R}^2 bezüglich der 1-Norm

ist. Seien also 3 Stopps des LKWs gegeben durch $k_1 = (x_1, y_1)$, $k_2 = (x_2, y_2)$, $k_3 = (x_3, y_3)$ mit $d((x_1, y_1), (x_2, y_2)) = d((x_1, y_1), (x_3, y_3)) = 1$ und $d((x_2, y_2), (x_3, y_3)) = 1/2m$. Dann gilt

$$\begin{aligned}
 x_1 \geq x_2 \geq x_3 : & \quad 1 = d(k_1, k_3) = d(k_1, k_2) + d(k_2, k_3) = 1 + (1/2m) \\
 x_2 \geq x_1 \geq x_3 : & \quad 1/2m = d(k_2, k_3) = d(k_2, k_1) + d(k_1, k_3) = 1 + 1 \\
 x_3 \geq x_1 \geq x_2 : & \quad 1/2m = d(k_3, k_2) = d(k_3, k_1) + d(k_1, k_2) = 1 + 1 \\
 x_3 \geq x_2 \geq x_1 : & \quad 1 = d(k_3, k_1) = d(k_3, k_2) + d(k_2, k_1) = (1/2m) + 1 \\
 x_1 \geq x_3 \geq x_2 : & \quad 1 = d(k_1, k_2) = d(k_1, k_3) + d(k_3, k_2) = 1 + (1/2m) \\
 x_2 \geq x_3 \geq x_1 : & \quad 1 = d(k_2, k_1) = d(k_2, k_3) + d(k_3, k_1) = (1/2m) + 1
 \end{aligned}$$

Somit kann es keine 3 Stopps für den LKW geben, welche die Distanzen von (Tang et al., 2019) erfüllen. \square

Anmerkung 4.9.9. Die pseudopolynomielle Transformation in (Tang et al., 2019) für $p \in \{1, \infty\}$ im \mathbb{R}^2 funktioniert mit einigen Anpassungen. Wird für den LKW die Metrik mit $p \in \{1, \infty\}$ und für die Drohne die Metrik mit $p \notin \{1, \infty\}$ vorausgesetzt, so gilt weiterhin, dass solch eine Einbettung nicht möglich ist. Ein solches Vorliegen zweier verschiedener Metriken ist nicht unrealistisch, da die Drohne nicht an das Straßennetzwerk gebunden ist, jedoch aber der LKW. Nun wird eine pseudopolynomielle Transformation bereitgestellt, die die Idee aus (Tang et al., 2019) widerspiegelt und im \mathbb{R}^2 bezüglich der 1-Norm eingebettet ist. Sei eine 3-Partition Instanz I mit ganzzahligen Parametern $B > 3$, $m > 0$ und h_i mit $\frac{B}{4} < h_i < \frac{B}{2}$ für alle $i \in \{1, \dots, 3m\}$ gegeben. Außerdem gilt $\sum_{i \in \{1, \dots, 3m\}} h_i = Bm$. Mit $\Psi(I)$ wird die folgende DRS Instanz bezeichnet:

$$v^T = 1 \quad v^D = 16(m + 1)$$

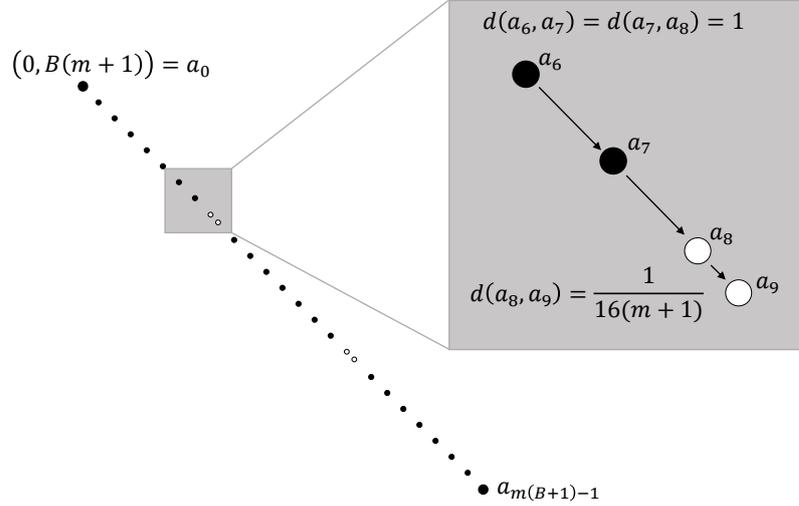


Abbildung 4.21: Kunden der DRS Instanz $\Psi(I)$ mit $m = 3$ und $B = 8$, die der LKW bedient

$$a_i = \left(\frac{i}{2} - \frac{16(m+1)-1}{32(m+1)} \left\lfloor \frac{i}{B+1} \right\rfloor, B(m+1) - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)} \left\lfloor \frac{i}{B+1} \right\rfloor \right)$$

$$i \in \{0, \dots, m(B+1)-1\}$$

$$b_j = (4h_j(m+1), B(m+1) + 4h_j(m+1)) \quad j \in \{0, \dots, 3m\}$$

$$s_k = \left(\frac{1}{4} + \frac{k(B+1)}{2} - \frac{16(m+1)-1}{32(m+1)}k + \frac{1}{128(m+1)}, \right.$$

$$\left. B(m+1) + \frac{1}{4} - \frac{k(B+1)}{2} + \frac{16(m+1)-1}{32(m+1)}k - \frac{1}{128(m+1)} \right)$$

$$k \in \{0, \dots, m-2\}$$

$$c = \left(4(m+1)mB - \frac{16(m+1)-1}{64(m+1)}(m-1) + \frac{m+m(B+1)-2}{4}, \right.$$

$$\left. (4mB+B)(m+1) + \frac{m}{4} + \frac{16(m+1)-1}{64(m+1)}(m-1) - \frac{m(B+1)}{4} \right).$$

Hierbei dient a_0 als Depot. Die LKW-Tour ist

$$(a_0, a_1, a_2, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{m(B+1)-1}, a_0).$$

Die Entscheidungsfrage lautet „Gibt es eine Lösung mit Zielfunktionswert ϕ , sodass $\phi \leq 2mB + \frac{2(m-1)}{16(m+1)}$ erfüllt ist?“. Die genauen Positionen der LKW-Kunden sind Abbildung 4.21 zu entnehmen. Zunächst wird gezeigt, dass die folgenden Gleichungen und Ungleichungen gelten:

$$d(a_i, a_{i-1}) = \frac{1}{16(m+1)} \quad i \in \{1, \dots, m(B+1) - 1\} \quad (4.53)$$

$$\wedge i \equiv 0 \pmod{B+1}$$

$$d(a_i, a_{i-1}) = 1 \quad i \in \{1, \dots, m(B+1) - 1\} \quad (4.54)$$

$$\wedge i \not\equiv 0 \pmod{B+1}$$

$$d(a_i, b_j) = 8h_j(m+1) \quad i \in \{0, \dots, m(B+1) - 1\}, \quad (4.55)$$

$$j \in \{1, \dots, 3m\}$$

$$d(s_k, a_{(k+1)(B+1)-1}) = \frac{1}{2} \quad k \in \{0, \dots, m-2\} \quad (4.56)$$

$$d(s_k, a_{(k+1)(B+1)}) = \frac{1}{2} \quad k \in \{0, \dots, m-2\} \quad (4.57)$$

$$d(s_k, a_l) > \frac{1}{2} \quad k \in \{0, \dots, m-2\}, l \in A_k \quad (4.58)$$

$$d(c, a_i) = 8(m+1)mB + \frac{m-1}{2} \quad i \in \{0, \dots, m(B+1) - 1\} \quad (4.59)$$

mit $A_k := \{0, \dots, m(B+1) - 1\} \setminus \{(k+1)(B+1), (k+1)(B+1) - 1\}$. Nun wird die Aussage (4.53) bewiesen. Sei $i = k(B+1)$. Dann ist

$$\begin{aligned} d(a_i, a_{i-1}) &= \left(\frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k - \frac{(i-1)}{2} + \frac{16(m+1)-1}{32(m+1)}(k-1) \right) \\ &\quad + \left(-\frac{(i-1)}{2} + \frac{16(m+1)-1}{32(m+1)}(k-1) + \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k \right) \\ &= \frac{1}{2} - \frac{16(m+1)-1}{32(m+1)} + \frac{1}{2} - \frac{16(m+1)-1}{32(m+1)} = \frac{1}{16(m+1)}. \end{aligned}$$

Sei nun $i = k(B+1) + l$ mit $l \in [1, \dots, B]$. Dann ist

$$\begin{aligned} d(a_i, a_{i-1}) &= \left(\frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k - \frac{(i-1)}{2} + \frac{16(m+1)-1}{32(m+1)}k \right) \\ &\quad + \left(-\frac{(i-1)}{2} + \frac{16(m+1)-1}{32(m+1)}k + \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k \right) = 1. \end{aligned}$$

Somit gilt auch (4.54). Seien $i = k(B+1) + l$ mit $l \in [0, \dots, B]$ und $j \in \{1, \dots, 3m\}$ gegeben. Dann ist

$$\begin{aligned} d(a_i, b_j) &= \left(4h_j(m+1) - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)}k \right) \\ &\quad + \left(4h_j(m+1) + B(m+1) - B(m+1) + \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k \right) \\ &= 8h_j(m+1). \end{aligned}$$

Somit folgt (4.55). Seien

$$a_i(x) := \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)} \left\lfloor \frac{i}{B+1} \right\rfloor$$

$$\begin{aligned}
a_i(y) &:= B(m+1) - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)} \left\lfloor \frac{i}{B+1} \right\rfloor \\
s_k(x) &:= \frac{1}{4} + \frac{k(B+1)}{2} - \frac{16(m+1)-1}{32(m+1)}k + \frac{1}{128(m+1)} \\
s_k(y) &:= B(m+1) - \frac{1}{4} + \frac{k(B+1)}{2} + \frac{16(m+1)-1}{32(m+1)}k - \frac{1}{128(m+1)}.
\end{aligned}$$

Jetzt werden folgende Aussagen für $i = (k+1)(B+1) - 1$ bewiesen:

$$a_{i+1}(x) \leq s_k(x) \leq a_{i+2} \quad (4.60)$$

$$a_i(y) \leq s_k(y) \leq a_{i-1}(y). \quad (4.61)$$

Die Positionen von $a_{i-1}, a_i, a_{i+1}, a_{i+2}$ und s_k im \mathbb{R}^2 sind der Abbildung 4.22 zu entnehmen. Es gilt

$$\begin{aligned}
s_k(x) - a_{i+1}(x) &= \frac{1}{4} + \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k + \frac{1}{128(m+1)} - \frac{i+1}{2} \\
&\quad + \frac{16(m+1)-1}{32(m+1)}(k+1) = \frac{1}{4} - \frac{3}{128(m+1)} > 0
\end{aligned}$$

$$\begin{aligned}
a_{i+2}(x) - s_k(x) &= \frac{i+2}{2} - \frac{16(m+1)-1}{32(m+1)}(k+1) - \frac{1}{4} - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)}k \\
&\quad - \frac{1}{128(m+1)} \\
&= \frac{1}{4} + \frac{3}{128(m+1)} > 0,
\end{aligned}$$

womit (4.60) folgt. (4.61) ergibt sich aus

$$\begin{aligned}
s_k(y) - a_i(y) &= B(m+1) + \frac{1}{4} - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)}k - \frac{1}{128(m+1)} \\
&\quad - B(m+1) + \frac{1}{2} - \frac{16(m+1)-1}{32(m+1)}k = \frac{1}{4} - \frac{1}{128(m+1)} > 0
\end{aligned}$$

$$\begin{aligned}
a_{i-1}(y) - s_k(y) &= B(m+1) - \frac{i-1}{2} + \frac{16(m+1)-1}{32(m+1)}k - B(m+1) - \frac{1}{4} \\
&\quad + \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)}k + \frac{1}{128(m+1)} = \frac{1}{4} + \frac{1}{128(m+1)} \\
&> 0.
\end{aligned}$$

Sei nun $i = (k+1)(B+1) - 1$. Dann ist

$$\begin{aligned}
d(s_k, a_i) &= \left(\frac{1}{4} + \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k + \frac{1}{128(m+1)} - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)}k \right) \\
&\quad + \left(B(m+1) + \frac{1}{4} - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)}k - \frac{1}{128(m+1)} \right)
\end{aligned}$$

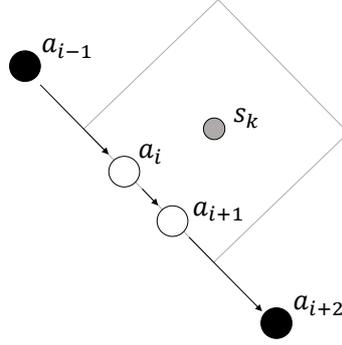


Abbildung 4.22: Positionen der LKW-Kunden $a_{i-1}, a_i, a_{i+1}, a_{i+2}$ mit $i = (k + 1)(B + 1) - 1$ sowie die Position des Drohnen-Kunden s_k . Die grau gekennzeichnete Menge entspricht der Sphäre mit Mittelpunkt s_k und Radius $\frac{1}{2}$.

$$-B(m+1) + \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k = \frac{1}{2}$$

und

$$\begin{aligned} d(s_k, a_{i+1}) &= \left(\frac{1}{4} + \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k + \frac{1}{128(m+1)} \right. \\ &\quad \left. - \frac{(i+1)}{2} + \frac{16(m+1)-1}{32(m+1)}(k+1) \right) + \left(B(m+1) + \frac{1}{4} \right. \\ &\quad \left. - \frac{(i+1)}{2} + \frac{16(m+1)-1}{32(m+1)}k - \frac{1}{128(m+1)} - B(m+1) \right. \\ &\quad \left. + \frac{(i+1)}{2} - \frac{16(m+1)-1}{32(m+1)}(k+1) \right) = \frac{1}{2}. \end{aligned}$$

Es gelten also die Distanzen (4.56) und (4.57). Sei $j < i = (k + 1)(B + 1) - 1$. Es gilt $\frac{i+1}{B+1} - 1 = k$.

$$\begin{aligned} d(s_k, a_j) &= \left(\frac{1}{4} + \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k + \frac{1}{128(m+1)} - \frac{j}{2} \right. \\ &\quad \left. + \frac{16(m+1)-1}{32(m+1)} \left\lfloor \frac{j}{B+1} \right\rfloor \right) + \left(B(m+1) - \frac{j}{2} \right. \\ &\quad \left. + \frac{16(m+1)-1}{32(m+1)} \left\lfloor \frac{j}{B+1} \right\rfloor - B(m+1) - \frac{1}{4} + \frac{i}{2} \right. \\ &\quad \left. - \frac{16(m+1)-1}{32(m+1)}k + \frac{1}{128(m+1)} \right) \\ &= (i-j) + \frac{1}{64(m+1)} + \frac{16(m+1)-1}{16(m+1)} \left(\left\lfloor \frac{j}{B+1} \right\rfloor - k \right) \\ &\geq (i-j) + \frac{1}{64(m+1)} + \frac{16(m+1)-1}{16(m+1)} \left(\frac{j}{B+1} - 1 - \frac{i+1}{B+1} + 1 \right) \end{aligned}$$

Da $B > 3$ ist, gilt

$$\begin{aligned}
& 32B(m+1)(2(i-j)-1) + B+1 - 96(m+1) + 4(i-j) + 1 > 0 \\
& \Leftrightarrow 2(i-j)B + (B+1+4(i-j)-64(m+1)+4) \frac{1}{32(m+1)} > B+1 \\
& \Leftrightarrow 2(i-j) \left(\frac{B}{B+1} \right) + \frac{B+1+4(i-j)-64(m+1)+4}{32(m+1)(B+1)} > 1 \\
& \Leftrightarrow (i-j) + \frac{1}{64(m+1)} + \frac{16(m+1)-1}{16(m+1)} \left(\frac{j-i}{B+1} - \frac{1}{B+1} \right) > \frac{1}{2}.
\end{aligned}$$

Somit ist

$$d(s_k, a_j) \geq (i-j) + \frac{1}{64(m+1)} + \frac{16(m+1)-1}{16(m+1)} \left(\frac{j}{B+1} - 1 - \frac{i+1}{B+1} + 1 \right) > \frac{1}{2}.$$

Sei nun $j > i$ mit $i = (k+1)(B+1)$. Die Distanz zwischen s_k und a_j wird nun abgeschätzt:

$$\begin{aligned}
d(s_k, a_j) &= \frac{j}{2} - \frac{16(m+1)-1}{32(m+1)} \left\lfloor \frac{j}{B+1} \right\rfloor - \frac{1}{4} - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)} k - \frac{1}{128(m+1)} \\
&\quad + \left(B(m+1) + \frac{1}{4} - \frac{i}{2} + \frac{16(m+1)-1}{32(m+1)} k - \frac{1}{128(m+1)} - B(m+1) \right) \\
&\quad + \frac{j}{2} - \frac{16(m+1)-1}{32(m+1)} \left\lfloor \frac{j}{B+1} \right\rfloor \\
&\geq (j-i) - \frac{16(m+1)-1}{16(m+1)} \left(\frac{j}{B+1} \right) + \frac{16(m+1)-1}{16(m+1)} k - \frac{1}{64(m+1)} \\
&= (j-i) + \frac{16(m+1)-1}{16(m+1)} \left(\frac{i+1}{B+1} - 1 - \frac{j}{B+1} \right) - \frac{1}{64(m+1)} \\
&= (j-i) + \frac{16(m+1)-1}{16(m+1)} \left(\frac{i-j-B}{B+1} \right) - \frac{1}{64(m+1)} \\
&= (j-i) \left(1 - \frac{16(m+1)-1}{16(m+1)(B+1)} \right) - \frac{B(16(m+1)-1)}{16(m+1)(B+1)} - \frac{1}{64(m+1)} \\
&= (j-i) \left(\frac{64(m+1)(B+1) - 64(m+1) - 4}{64(m+1)(B+1)} \right) \\
&\quad - \frac{B(64(m+1)-4) + B+1}{64(m+1)(B+1)}.
\end{aligned}$$

Für $B > 3$ gilt

$$\begin{aligned}
& 32(m+1)B + 3B - 9 - 32(m+1) > 0 \\
& \Leftrightarrow 128(m+1)B - 8 - 96(m+1)B + 4B - B - 1 - 32(m+1) > 0.
\end{aligned}$$

Da $j-i \geq 2$ erfüllt ist, gilt

$$\begin{aligned}
& 128(m+1)B - 8 - 96(m+1)B + 4B - B - 1 - 32(m+1) \\
& < (j-i)(64(m+1)B - 4) - B(96(m+1) - 4) - B - 1 - 32(m+1).
\end{aligned}$$

Somit ist

$$\begin{aligned}
& (j-i)(64(m+1)B-4) - B(96(m+1)-4) - B-1 - 32(m+1) > 0 \\
\Leftrightarrow & (j-i)(64(m+1)(B+1) - 64(m+1) - 4) - B(64(m+1) - 4) - B-1 \\
& > 32(m+1)(B+1) \\
\Leftrightarrow & (j-1) \left(\frac{64(m+1)(B+1) - 64(m+1) - 4}{64(m+1)(B+1)} \right) - \frac{B(64(m+1) - 4) + B+1}{64(m+1)(B+1)} \\
& > \frac{1}{2}.
\end{aligned}$$

Insgesamt folgt

$$d(s_k, a_j) > \frac{1}{2}$$

und somit (4.58).

Nun wird gezeigt, dass der Abstand zwischen c und a_i für alle $i \in \{0, \dots, m(B+1)-1\}$ genau $16(m+1)mB+m-1$ ist. Sei dazu $i = k(B+1)+l$ mit $l < B+1$. Dann ist

$$\begin{aligned}
d(c, a_i) &= 4(m+1)mB - \frac{16(m+1)-1}{64(m+1)}(m-1) + \frac{m+m(B+1)-2}{4} \\
& - \frac{i}{2} - \frac{16(m+1)-1}{32(m+1)}k + \left((4mB+B)(m+1) + \frac{m}{4} \right. \\
& + \frac{16(m+1)-1}{64(m+1)}(m-1) - \frac{1}{4}m(B+1) - B(m+1) + \frac{i}{2} \\
& \left. + \frac{16(m+1)-1}{32(m+1)}k \right) \\
& = 8(m+1)mB + \frac{2m}{4} - \frac{2}{4} = 8(m+1)mB + \frac{m-1}{2}
\end{aligned}$$

Somit gilt auch (4.59).

Mit (4.53) und (4.54) lässt sich nun eine untere Schranke $US(LKW)$ für die Reisezeit des LKW ermitteln:

$$\begin{aligned}
US(LKW) &= mB + \frac{(m-1)}{16(m+1)} + d(a_0, a_{m(B+1)-1}) \\
&= mB + \frac{(m-1)}{16(m+1)} + \left(\frac{m(B+1)-1}{2} - \frac{16(m+1)-1}{32(m+1)}(m-1) - 0 \right) \\
& + \left(B(m+1) - B(m+1) + \frac{m(B+1)-1}{2} \right. \\
& \left. - \frac{16(m+1)-1}{32(m+1)}(m-1) \right) \\
&= mB + \frac{(m-1)}{16(m+1)} + m(B+1) - 1 - \frac{16(m+1)-1}{16(m+1)}(m-1) \\
&= 2mB + \frac{2(m-1)}{16(m+1)}.
\end{aligned}$$

Diese Schranke wird genau dann erreicht, wenn der LKW ohne Pausen seine Kunden nacheinander bedient. Nun wird eine untere Schranke $US(DR)$ für die Reisezeit der Drohne bestimmt:

$$\begin{aligned} US(DR) &= \frac{2}{v^D} \left(\sum_{j=1}^{3m} 8h_j(m+1) \right) + \frac{2(m-1)}{2v^D} + \frac{2}{v^D} \cdot \left(8(m+1)mB + \frac{m-1}{2} \right) \\ &= \frac{16mB(m+1)}{16(m+1)} + \frac{(m-1)}{16(m+1)} + \frac{2}{16(m+1)} \cdot \left(8(m+1)mB + \frac{m-1}{2} \right) \\ &= mB + \frac{(m-1)}{16(m+1)} + mB + \frac{(m-1)}{16(m+1)} = 2mB + \frac{2(m-1)}{16(m+1)}. \end{aligned}$$

Diese Schranke wird genau dann erreicht, wenn die Drohne ohne Wartezeiten alle ihre Kunden bedient und sie bei $a_{(k+1)(B+1)-1}$ gestartet wird, um s_k zu bedienen. Dies muss für alle $k \in \{0, \dots, m-2\}$ erfüllt sein.

Sei eine 3-Partition-Instanz I gegeben, sodass es m viele disjunkte Teilmengen A_0, A_2, \dots, A_{m-1} gibt mit $A_l = \{h_{\sigma_l(1)}, h_{\sigma_l(2)}, h_{\sigma_l(3)}\} \subseteq \{h_1, \dots, h_{3m}\}$ und $h_{\sigma_l(1)} + h_{\sigma_l(2)} + h_{\sigma_l(3)} = B$ für alle $l \in \{0, \dots, m-1\}$. Da $US(LKW) = US(DR) = 2mB + \frac{2(m-1)}{16(m+1)}$ für $\Psi(I)$ gilt, muss der LKW ohne Pausen nacheinander seine Kunden bedienen. Die Drohne muss jeweils bei $a_{(k+1)(B+1)-1}$ für alle $k \in \{0, \dots, m-2\}$ gestartet werden. Wird sie bei $a_{l(B+1)}$ gestartet, um nacheinander $b_{\sigma_l(1)}, b_{\sigma_l(2)}, b_{\sigma_l(3)}$ zu bedienen, so bedient sie zunächst Kunde $b_{\sigma_l(1)}$ zum Zeitpunkt

$$\frac{d(a_{l(B+1)}, b_{\sigma_l(1)})}{v^D} = \frac{8h_{\sigma_l(1)}(m+1)}{16(m+1)} = \frac{h_{\sigma_l(1)}}{2}.$$

Die gleiche Zeit wird benötigt, um zurück zu einem der Kunden zu fliegen, der durch den LKW bedient wird. Daher begegnen sich Drohne und LKW ohne Wartezeiten bei $a_{l(B+1)+h_{\sigma_l(1)}}$. Nachdem die Drohne $b_{\sigma_l(1)}, b_{\sigma_l(2)}, b_{\sigma_l(3)}$ bedient hat, begegnen sich Drohne und LKW ohne Wartezeiten bei

$$a_{l(B+1)+h_{\sigma_l(1)+h_{\sigma_l(2)+h_{\sigma_l(3)}}} = a_{l(B+1)+B} = a_{l(B+1)+(B+1)-1} = a_{(l+1)(B+1)-1}.$$

Die LKW-Tour und die Drohnen-Tour ist Abbildung 4.23 zu entnehmen. Nachdem die Drohne die Kunden b_j für alle $j \in \{1, \dots, 3m\}$ und die Kunden s_k für alle $k \in \{0, \dots, m-2\}$ bedient hat, begegnet sie dem LKW zum Zeitpunkt $\frac{(m-1)}{16(m+1)} + mB$ am Ort $a_m(B+1) - 1$. Anschließend bedient die Drohne den Kunden c . Zum Zeitpunkt $\frac{2(m-1)}{16(m+1)} + 2mB$ begegnet sie dem LKW am Ort a_0 . Diese Lösung führt zu einer *Ja*-Antwort.

Sei nun eine 3-Partition-Instanz I gegeben, wobei $\Psi(I)$ mit *Ja* beantwortet wird. Dies bedeutet, dass der LKW ohne Unterbrechung seine Kunden bedient und die Drohne ohne Unterbrechung ihre Kunden bedient. Ferner muss die Drohne bei $a_{(k+1)(B+1)-1}$ für alle $k \in \{0, \dots, m-2\}$ gestartet werden, um s_k zu bedienen. Wird die Drohne bei a_0 gestartet, um Kunde c zu bedienen, so kann sie erst bei $a_{m(B+1)-1}$ erneut gestartet werden. Ab diesem Punkt ist es ihr jedoch nur noch möglich, einen Kunden zu bedienen, da der LKW ausschließlich das Depot ansteuert. Wird die Drohne bei a_i mit $i \in \{1, \dots, m(B+1)-1\}$ gestartet, so begegnen sich Drohne und LKW erst wieder am Depot a_0 . Folglich wird jede

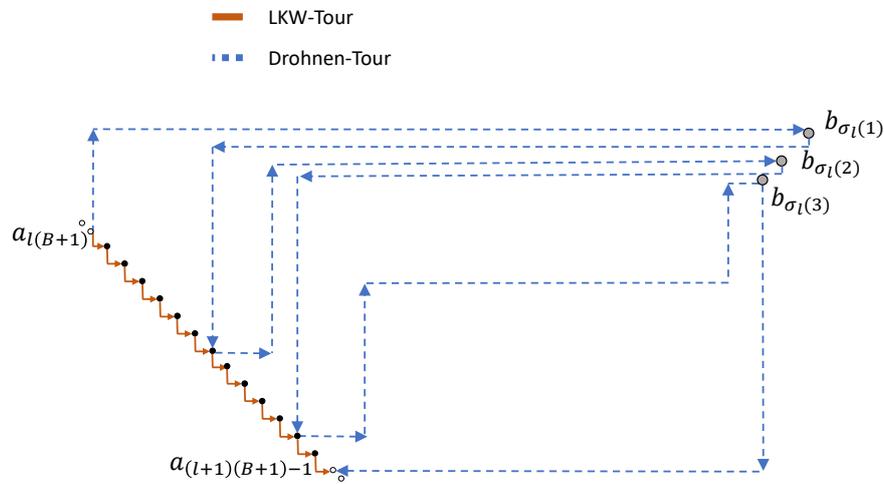


Abbildung 4.23: LKW-Tour und Drohnen-Tour für ein Zeitsegment der Länge $B = 14$ und $h_{\sigma_l(1)} = 7, h_{\sigma_l(2)} = 5, h_{\sigma_l(3)} = 2$.

Lösung durch eine Lösung dominiert, bei der die Drohne bei $a_{m(B+1)-1}$ gestartet wird, um den Kunden c zu bedienen. Somit verbleibt für die übrigen Kunden, die durch die Drohne bedient werden, das Zeitintervall $[0, mB + \frac{(m-1)}{16(m+1)}]$. Dieses setzt sich wiederum aus m Zeitintervallen der Länge B zusammen, die jeweils getrennt sind, da die Drohne nach jedem k -ten Abschnitt Kunde s_k bedienen muss, um $US(DR)$ als Reisezeit einhalten zu können.

Kapitel 5

Abschluss und Ausblick

In der vorliegenden Arbeit wurde das Line-TSPTW um eine Drohne erweitert und so das Line-TSPwD konstruiert. Zunächst wurde das Line-TSPwD als MIP modelliert und ein CP Ansatz vorgestellt. Im Hauptteil dieser Arbeit wurde die Komplexität des Line-TSPwD in Abhängigkeit verschiedener Bedingungen untersucht. Zunächst wurde der kontinuierliche Fall betrachtet. Dort wurde zwischen den Verhältnis der Drohnengeschwindigkeit zur Truckgeschwindigkeit und dem Vorhandensein von Release Dates bzw. Deadlines unterschieden. Zum Abschluss wurden die Ergebnisse noch auf den diskreten Fall übertragen. Diese Ergebnisse lassen sich auf andere in der Literatur bekannte Probleme anwenden. Es wurde gezeigt, dass das TSP-D (Agatz et al., 2018), das FSTSP (Murray & Chu, 2015), das DRS (Tang et al., 2019) und das DSP (Boysen et al., 2018) \mathcal{NP} -schwer im strengen Sinne sind, selbst wenn alle Kunden auf einer Linie liegen. Das heißt, dass die Positionen der Kunden im \mathbb{R}^1 liegen und die Minkowski-Distanz für ein beliebiges $1 \leq p \leq \infty$ verwendet wird.

Für den kontinuierlichen Fall, in dem keine Zeitfenster vorhanden sind und die Drohne schneller ist als der LKW, ist noch offen, ob das Problem \mathcal{NP} -schwer im strengen Sinne ist oder ob ein pseudopolynomieller Algorithmus existiert, der das Problem löst.

Da die vorliegende Arbeit ein MIP und einen CP-Ansatz liefert, würde es sich anbieten, die gegebenen Modelle miteinander zu vergleichen. Die Auswirkungen verschiedener Instanzstrukturen auf die Rechenzeit und den Zielfunktionswert könnten z.B. durch Variation der Zeitfenstergröße oder des Geschwindigkeitsverhältnisses getestet werden. Da nun der Komplexitätsstatus für den diskreten Fall geklärt ist und sowohl ein polynomieller als auch ein pseudopolynomieller Ansatz ausgeschlossen sind, ist die Konstruktion eines B&B-Verfahrens denkbar.

Eine weitere Möglichkeit wäre auch, die Berücksichtigung anderer Zielfunktionen, wie zum Beispiel die Minimierung der Gesamtwartezeiten der Kunden, also entsprechend die Erweiterung des Line-TRP, oder die Minimierung von CO_2 -Emissionen als Ziel.

Literatur

- Aaronson, S. (2016). P=?NP. In J. Nash & M. Rassias (Hrsg.), *Open problems in mathematics* (S. 1-121). Springer. Zugriff auf <https://doi.org/10.1007/978-3-319-32162-2> (ISBN:978-3-319-32160-8)
- Agatz, N., Bouman, P. & Schmidt, M. (2018, August). Optimization Approaches for the Traveling Salesman Problem with Drone. *Transportation Science*, 52 (4), 965-981. Zugriff auf <https://doi.org/10.1287/trsc.2017.0791>
- Amazon. (2018). *Prime Air*. Zugriff am 20/05/2021 auf <https://www.aboutamazon.de/innovationen/prime-air>
- BMVI. (2020). *Unbemannte Luftfahrtsysteme und innovative Luftfahrtkonzepte - Aktionsplan der Bundesregierung*. Zugriff am 20/05/2021 auf https://www.bmvi.de/SharedDocs/DE/Anlage/DG/aktionsplan-drohnen.pdf?__blob=publicationFile
- Bock, S. (2024). Vehicle routing for connected service areas - a versatile approach covering single, hierarchical, and bi-criteria objectives. *European Journal of Operational Research*, 313 (3), 905-925. Zugriff auf <https://doi.org/10.1016/j.ejor.2023.08.051>
- Bock, S. & Janiszczak, K. (o. J.). *Complexity of the Line-TSP with a drone*. Bergische Universität Wuppertal. (unveröffentlicht)
- Bock, S. & Klamroth, K. (2013). Minimizing sequence-dependent setup costs in feeding batch processes under due date restrictions. *Journal of Scheduling*, 16 (5), 479-494. Zugriff auf <https://doi.org/10.1007/s10951-013-0334-0>
- Bouman, P., Agatz, N. & Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 72 (4), 528-542. Zugriff auf <https://doi.org/10.1002/net.21864>
- Boysen, N., Briskorn, D., Fedtke, S. & Schwerdfeger, S. (2018, 10). Drone delivery from trucks: Drone scheduling for given truck routes. *Networks*, 72, 506-527. Zugriff auf <https://doi.org/10.1002/net.21847>
- Boysen, N., Fedtke, S. & Schwerdfeger, S. (2021, 03). Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, 43, 1-58. Zugriff auf <https://doi.org/10.1007/s00291-020-00607-8>
- Burns, S. (2017). *Drone meets delivery truck*. Zugriff am 25/05/2021 auf <https://www.ups.com/us/es/services/knowledge-center/article.page?kid=cd18bdc2&articlesource=longitudes>
- Cheng, T. & Ding, Q. (1998). The complexity of scheduling starting time dependent tasks with release times. *Information Processing Letters*, 65 (2), 75-79. Zugriff auf [https://doi.org/10.1016/S0020-0190\(97\)00195-6](https://doi.org/10.1016/S0020-0190(97)00195-6)
- Cheng, T. & Ding, Q. (1999). The time dependent machine makespan problem is strongly NP-complete. *Computers & Operations Research*, 26 (8), 749-754. Zugriff auf [https://doi.org/10.1016/S0305-0548\(98\)00093-8](https://doi.org/10.1016/S0305-0548(98)00093-8)
- Chung, S. H., Sah, B. & Lee, J. (2020). Optimization for drone and drone-truck combined operations: A review of the state of the art and

- future directions. *Computers & Operations Research*, 123, 105004.
Zugriff auf <https://doi.org/10.1016/j.cor.2020.105004>
- Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of the third annual acm symposium on theory of computing* (S. 151–158). New York, NY, USA: Association for Computing Machinery. Zugriff auf <https://doi.org/10.1145/800157.805047> (ISBN:9781450374644)
- Dalamagkidis, K. (2015). Classification of UAVs. In G. J. V. Kimon P. Valavanis (Hrsg.), *Handbook of Unmanned Aerial Vehicles*. Springer, Dordrecht. Zugriff auf https://doi.org/10.1007/978-90-481-9707-1_92 (ISBN:978-90-481-9706-4)
- Dell'Amico, M., Montemanni, R. & Novellani, S. (2021). Algorithms based on branch and bound for the flying sidekick traveling salesman problem. *Omega*, 104, 102493. Zugriff auf <https://doi.org/10.1016/j.omega.2021.102493>
- Garey, M. R. & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)* (1. Aufl.). W. H. Freeman. (ISBN:0716710455)
- Garey, M. R., Tarjan, R. E. & Wilfong, G. T. (1988). One-Processor Scheduling with Symmetric Earliness and Tardiness Penalties. *Mathematics of Operations Research*, 13 (2), 330-348. Zugriff auf <https://doi.org/10.1287/moor.13.2.330>
- Golden, B., Raghavan, S. & Wasil, E. (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges* (Bd. 43). Springer New York, NY. Zugriff auf <https://doi.org/10.1007/978-0-387-77778-8> (ISBN:978-0-387-77777-1)
- Gonzalez-R, P. L., Canca, D., Andrade-Pineda, J. L., Calle, M. & Leon-Blanco, J. M. (2020). Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transportation Research Part C: Emerging Technologies*, 114, 657-680. Zugriff auf <https://doi.org/10.1016/j.trc.2020.02.030>
- Graham, R., Lawler, E., Lenstra, J. & Kan, A. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. In P. Hammer, E. Johnson & B. Korte (Hrsg.), *Discrete Optimization II* (Bd. 5, S. 287-326). Elsevier. Zugriff auf [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X)
- Gupta, J. N. & Gupta, S. K. (1988). Single facility scheduling with nonlinear processing times. *Computers & Industrial Engineering*, 14 (4), 387-393. Zugriff auf [https://doi.org/10.1016/0360-8352\(88\)90041-1](https://doi.org/10.1016/0360-8352(88)90041-1)
- International Civil Aviation Organization. (2011). *Unmanned Aircraft Systems (UAS)*. Zugriff am 21/05/2021 auf <https://www.icao.int/Meetings/UAS/Documents/Circular%20328.en.pdf>
- Karmarkar, N. (1984). A New Polynomial-Time Algorithm for Linear Programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing* (S. 302–311). New York, NY, USA: Association for Computing Machinery. Zugriff auf <https://doi.org/10.1145/800057.808695> (ISBN:0897911334)
- Karp, R. M. (1972). Reducibility among Combinatorial Problems. In R. E. Miller, J. W. Thatcher & J. D. Bohlinger (Hrsg.), *Complexity of*

- computer computations: Proceedings of a symposium on the complexity of computer computations* (S. 85–103). Boston, MA: Springer US. Zugriff auf https://doi.org/10.1007/978-1-4684-2001-2_9 (ISBN:978-1-4684-2001-2)
- Kaufmann, J. (2019). *Dieses Startup liefert Medikamente per Drohne — und hat bereits tausende Menschenleben gerettet*. Businessinsider. Zugriff am 20/05/2021 auf <https://www.businessinsider.de/tech/dieses-startup-liefert-medikamente-per-drohne-und-hat-bereits-14000-menschenleben-gerettet-2019-5/>
- Koenen, J. (2021). „Gamechanger“ der Lieferdrohnen – So sieht das neue Modell von Wingcopter aus. Zugriff am 25/05/2021 auf <https://www.handelsblatt.com/unternehmen/handel-konsumgueter/logistik-gamechanger-der-lieferdrohnen-so-sieht-das-neue-modell-von-wingcopter-aus/27133194.html?ticket=ST-3799353-a5F3A9XLuGyBDU2Xb3Ml-ap1>
- Laborie, P., Rogerie, J., Shaw, P. & Vilím, P. (2018). IBM ILOG CP optimizer for scheduling. *Constraints*, 23, 210-250. Zugriff auf <https://doi.org/10.1007/s10601-018-9281-x>
- Lavor, C., Liberti, L., Lodwick, W. A. & Mendonça da Costa, T. (2017). The distance geometry problem (dgp). In *An introduction to distance geometry applied to molecular geometry* (S. 5–12). Cham: Springer International Publishing. Zugriff auf https://doi.org/10.1007/978-3-319-57183-6_2 (ISBN:978-3-319-57183-6)
- Little, A. D. (2018). *Prognostiziertes Aufkommen von urbanen Gütertransportleistungen weltweit in den Jahren 2010 bis 2050 (in Billionen Tonnenkilometern)*. Statista. Zugriff am 10/06/2024 auf <https://de.statista.com/statistik/daten/studie/1084712/umfrage/prognose-der-nachfrage-nach-urbanen-guetertransportleistungen-weltweit/>
- Matai, R., Singh, S. & Mittal, M. L. (2010). Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches. In D. Davendra (Hrsg.), *Traveling salesman problem* (Kap. 1). Rijeka: IntechOpen. Zugriff auf <https://doi.org/10.5772/12909> (ISBN:978-953-307-426-9)
- Murray, C. C. & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86-109. Zugriff auf <https://doi.org/10.1016/j.trc.2015.03.005>
- Otto, A., Agatz, N., Campbell, J., Golden, B. & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72 (4), 411-458. Zugriff auf <https://doi.org/10.1002/net.21818>
- Pharmazeutische Zeitung. (2021). *Wenn die Arzneimittel vom Himmel fallen*. Zugriff am 20/05/2021 auf <https://www.pharmazeutische-zeitung.de/wenn-die-arzneimittel-vom-himmel-fallen-124789/>
- Pinedo, M. (2008). *Scheduling: Theory, Algorithms, And Systems* (3. Aufl.). Springer. (ISBN:978-0387789347)
- Poikonen, S. & Golden, B. (2020). The Mothership and Drone Routing

- Problem. *INFORMS Journal on Computing*, 32 (2), 249-262. Zugriff auf <https://doi.org/10.1287/ijoc.2018.0879>
- Poodiack, R. D. & Wood, W. E. (2022). Geometry in the p-norm. In *Squigonometry: The study of imperfect circles* (S. 197–204). Cham: Springer International Publishing. Zugriff auf https://doi.org/10.1007/978-3-031-13783-9_18 (ISBN:978-3-031-13783-9)
- Psaraftis, H. N., Solomon, M. M., Magnanti, T. L. & Kim, T.-U. (1990). Routing and Scheduling on a Shoreline with Release Times. *Management Science*, 36 (2), 212-223. Zugriff auf <https://doi.org/10.1287/mnsc.36.2.212>
- Sedding, H. A. (2020). Scheduling jobs with a V-shaped time-dependent processing time. *Journal of Scheduling*, 24, 751–768. Zugriff auf <https://doi.org/10.1007/s10951-020-00665-4>
- Tang, Z., Hoeve, W.-J. v. & Shaw, P. (2019). A study on the traveling salesman problem with a drone. In L.-M. Rousseau & K. Stergiou (Hrsg.), *Integration of constraint programming, artificial intelligence, and operations research* (S. 557–564). Cham: Springer International Publishing. Zugriff auf https://doi.org/10.1007/978-3-030-19212-9_37
- Tittmann, P. (2019). *Einführung in die Kombinatorik* (3. Aufl.). Springer Spektrum Berlin, Heidelberg. Zugriff auf <https://doi.org/10.1007/978-3-662-58921-2> (ISBN:978-3-662-58920-5)
- Tsitsiklis, J. N. (1992). Special cases of traveling salesman and repairman problems with time windows. *Networks*, 22 (3), 263-282. Zugriff auf <https://doi.org/10.1002/net.3230220305>
- Wegener, I. (2005). *Theoretische Informatik - eine algorithmenorientierte Einführung* (3. Aufl.). Vieweg+Teubner Verlag Wiesbaden. Zugriff auf <https://doi.org/10.1007/978-3-322-82204-8> (ISBN:978-3-8351-0033-6)
- Wiles, A. (1995). Modular Elliptic Curves and Fermat's Last Theorem. *Annals of Mathematics*, 141 (3), 443–551. Zugriff auf <https://doi.org/10.2307/2118559>
- Zipline. (o. J.). *Zipline*. Zugriff am 25/05/2021 auf <https://robots.ieee.org/robots/zipline/>