Antizipierende und faire Transportplanung

Christian Billing

Angaben zur Veröffentlichung / Publication details:

Billing, Christian. 2020. "Antizipierende und faire Transportplanung." Augsburg: Universität Augsburg.

Antizipierende und faire Transportplanung

Kumulative Dissertation
an der Wirtschaftswissenschaftlichen Fakultät
der Universität Augsburg
zur Erlangung des Grades eines
Doktors der Wirtschaftswissenschaften
(Dr. rer. pol.)

vorgelegt von

Christian Billing, M.Sc.

Erstgutachter:	Prof. Dr. Florian Jaehn
Zweitgutachter:	Prof. Dr. Robert Klein
Drittgutachter:	Prof. Dr. Axel Tuma
Vorsitzender der mündlichen Prüfung:	Prof. Dr. Jens O. Brunner
Datum der mündlichen Prüfung:	24.10.2018

Verzeichnis der Beiträge

Die Dissertation beinhaltet die folgenden Beiträge, welche zum Teil bei internationalen Journals eingereicht und zur Veröffentlichung angenommen wurden. Die jeweils angegebene Kategorie bezieht sich auf das Zeitschriftenranking JOURQUAL 3 des Verbands der Hochschullehrer für Betriebswirtschaft e.V. (VHB).

Beitrag B1:

Billing, C., Jaehn, F., and Wensing, T. (2018). A multiperiod auto-carrier transportation problem with probabilistic future demands. *Journal of Business Economics*, 88:1009-1028 (Kategorie B)
https://doi.org/10.1007/s11573-018-0904-x

Beitrag B2:

Billing, C. (2018). The multiperiod 2-customer vehicle routing problem.

Beitrag B3:

Pre-Print of: Billing, C., Jaehn, F., and Wensing, T. (2020). Fair Task Allocation Problem. *Annals of Operations Research*, 284:131-146 (Kategorie B)
https://doi.org/10.1007/s10479-018-3052-3

Inhaltsverzeichnis

I Einleitung . . . 1

1 Motivation . . . 1

2 Kurzbeschreibung der Beiträge . . . 6

II Beitrag B1: A multiperiod auto-carrier transportation problem with probabilistic future demands . . . 14

III Beitrag B2: The multiperiod 2-customer vehicle routing problem . . . 15

IV Beitrag B3: Fair Task Allocation Problem . . . 45

V Fazit und Ausblick . . . 74

Anmerkung: Die Nummerierung von Abbildungen, Tabellen und mathematischen Gleichungen erfolgt fortlaufend innerhalb der Kapitel. Ein Literaturverzeichnis befindet sich am Ende jedes Kapitels.

I Einleitung

1 Motivation

Die weltweite Automobilindustrie befindet sich seit Jahren in einem stetigen Aufschwung. So wurden im Jahr 2016 ca. 95 Millionen Fahrzeuge hergestellt, mehr als je zuvor. Verglichen mit ca. 79,9 Millionen produzierten Fahrzeugen im Jahr 2011 bedeutet dies einen Anstieg um fast 19% (OICA, 2018a), wobei der treibende Faktor für diese Entwicklung zweifelsohne der rasant wachsende Absatz in China ist. Dort stieg in diesem Zeitraum die Zahl der Neuzulassungen pro Jahr von 18,5 auf über 28 Millionen Fahrzeuge an (OICA, 2018b). Doch auch in traditionellen Märkten wie Europa und den USA kann von nachlassenden Absatzzahlen keine Rede sein. Dies ist durchaus bemerkenswert, da sich moderne Formen der Mobilität im Aufwind befinden. Zu nennen wären beispielsweise Car-Sharing Angebote, deren Nutzerzahl sich in Deutschland seit 2011 mehr als verachtfacht hat (Bundesverband CarSharing, oA). Ebenso werden die fahrscheinlose und kostenfreie Nutzung des öffentlichen Personennahverkehrs diskutiert. Das bekannteste derartige Projekt ist in der estnischen Hauptstadt Tallinn angesiedelt, welches jedoch laut einer Untersuchung von Cats et al. (2017) nicht zu einem Rückgang des Autoverkehrs geführt hat. Insgesamt ist festzuhalten, dass die Automobilindustrie ihren Umsatz auch in den kommenden Jahren weiter steigern wird. Eine Studie von McKinsey&Company (2016) kommt zu dem Ergebnis, dass zwar ein Großteil der erwarteten Umsatzsteigerung bis 2030 auf moderne Mobilitätsangebote zurückzuführen sein wird, jedoch auch der Umsatz aus Fahrzeugverkäufen weiter zulegen wird.

Obgleich sich das Absatzvolumen auf einem sehr hohen Niveau befindet, herrscht zwischen Automobilunternehmen ein großer Konkurrenzkampf. Laut Herold (2005, S. 6) führen die Globalisierung und die rasante Entwicklung der Informationstechnologie zu einer gesteigerten Markttransparenz. Dadurch werden Unterschiede bezüglich klassischer Wettbewerbsfaktoren wie Preis-Leistungs-Verhältnis oder Qualität abgebaut und der Faktor Kundenzufriedenheit gewinnt an Bedeutung. Gleichzeitig zeigt sich laut Dannenberg (2005, S. 40 f.), dass die Markentreue der Kunden nach und nach abnimmt. Allerdings kann die Loyalität der Kunden durch eine hohe Kundenzufriedenheit positiv beeinflusst werden, wie Skala-Gast (2012, S. 142 ff.) in einer breit angelegten Untersuchung feststellt. Elementare Kriterien für die Zufriedenheit des Kunden beim Kauf eines Autos sind dabei Lieferzeit und Liefertreue (vgl. Herold, 2005, S. 10 ff.). Ein bestelltes Fahrzeug sollte also möglichst schnell und zuverlässig an den Kunden geliefert werden. Dabei besteht enormes Verbesserungspotential, wie Herold (2005, S. 14 ff.) aufzeigt. Lieferzeiten von über acht Wochen und Lieferverzögerungen von über drei Wochen sind keine Seltenheit in Deutschland.

Somit fällt der Distributionslogistik, welche als "Bindeglied zwischen Produktion und Absatz" (Ihme, 2006, S. 347) fungiert, eine wesentliche Rolle im langfristigen Wettbewerb um den Kunden zu. Die zentrale Funktion der Distributionslogistik ist es, das fertige Produkt durch zeitliche und räumliche Transformation dem Kunden zur Verfügung zu stellen. Zur Erfüllung dieser Funktion müssen sowohl strategische Aufgaben wie die Lagernetzplanung und die Standortwahl der Lager als auch operative Aufgaben im Bestandsmanagement und Warentransport gelöst werden (vgl. Huber und Laverentz, 2012, S. 121 ff.). In der Automobilindustrie erfolgt der Vertrieb der Fahrzeuge in der Regel über Autohändler (vgl. Klug, 2010, S. 431), die somit in der Lieferkette als Auslieferungslager fungieren. Je nach Unternehmen können außerdem Zentral- und/oder Zwischenlager vorgeschaltet sein (vgl. Grafmüller, 2000, S. 102 f.). Der Transport kann per Schiff, Bahn und Lastkraftwagen (LKW) erfolgen, wobei letztere insbesondere für die Belieferung der Händler genutzt werden. Diese speziell für die Beladung mit Fahrzeugen konzipierten Transporter sind in der

Lage, zwischen sieben und elf Autos gleichzeitig zu befördern. Moderne Autotransporter verfügen über variable Ladeebenen, so dass unterschiedlichste Fahrzeugtypen transportiert werden können. Außerdem wird dadurch eine flexible Positionierung der geladenen Autos ermöglicht, wodurch Reloading-Prozesse auf Auslieferungstouren mit mehreren Händlern vermieden werden können (vgl. Klug, 2010, S. 435). Um Beschädigungen der Fahrzeuge während des Transports zu vermeiden, ist ein Trend von offenen hin zu geschlossenen Autotransportern zu beobachten (vgl. Ihme, 2006, S. 147).

Neben der bereits erwähnten Einhaltung von Lieferterminen ist bei der Fahrzeugdistribution auf geringe Kosten zu achten. Einen wesentlichen Beitrag dazu kann die kurzfristig ausgelegte Transportplanung leisten, die unter anderem die Optimierung von Auslieferungstouren zum Gegenstand hat (vgl. Fandel et al., 2009, S. 187). Die Basis solcher Fragestellungen bildet dabei das Vehicle-Routing-Problem (VRP), welches erstmals von Dantzig und Ramser (1959) erwähnt wurde. Bei diesem muss der Bedarf einer Menge an Kunden möglichst kostengünstig gedeckt werden, wofür Fahrzeuge mit beschränkter Kapazität zur Verfügung stehen, welche ihre Touren an einem Depot starten und abschließen. Mittlerweile existiert eine Vielzahl an Erweiterungen, oft motiviert durch praktische Fragestellungen. Beispielhaft zu nennen wären dabei die Existenz mehrerer Depots, die Berücksichtigung von Zeitfenstern oder die Unterscheidung zwischen Pick-Up-Kunden und Delivery-Kunden. Einen Überblick liefern Toth und Vigo (2002). Da in der Praxis Daten meist nicht bereits zu Beginn der Planung vollständig vorliegen oder zumindest mit Unsicherheit bedacht sind, ist die Untersuchung von dynamischen und stochastischen Varianten in den Fokus gerückt (vgl. Ritzinger et al., 2016). Bei der Auslieferung von Autos kommt erschwerend hinzu, dass - im Gegensatz zu vielen anderen Problemstellungen - die Kapazität eines Transporters nicht durch eine eindimensionale Größe wie *Anzahl*, *Gewicht* oder *Volumen* ausgedrückt werden kann. In der Literatur finden sich einige Arbeiten, die sich speziell mit der Optimierung von Touren für Autotransporter beschäftigen (vgl. Tadei et al. (2002), Dell'Amico et al. (2014), Cordeau et al. (2015) und Hu et al. (2015)).

Für die Optimierung der Tourenplanung, auch mit Blick auf eine höhere Kapazitätsauslastung, ist eine verbesserte Verknüpfung von Informationen zwischen Produktion und Absatz erforderlich, insbesondere bei einer make-to-order Produktion. Aktuell stehen dem Vertrieb Informationen über Zielorte und Übergabetermine erst nach Fertigstellung des Fahrzeugs durch die Produktion zur Verfügung, wie Werthmann et al. (2012) ausführen. Diesen "Schnittstellenbruch [...] bei der Übergabe des fertigen Fahrzeuges an die Distribution" stellte bereits Stautner (2001, S. 80) fest. Frühzeitige Informationen über zu erwartende Fertigstellungszeitpunkte von Fahrzeugen ermöglichen eine vorausschauende und effiziente Planung in der Distribution. Solche Informationen sind jedoch stets mit Unsicherheit behaftet, da in der Produktion kurzfristige Änderungen auftreten können. Auch bei einer make-to-stock Produktion ist der Zeitpunkt, zu dem ein Auto zur Auslieferung bereit steht, nicht präzise vorhersehbar. In diesem Fall besteht die Unsicherheit jedoch auf der Nachfrageseite, da Auftragseingänge durch die Händler stochastisch sind. Der erste Beitrag dieser kumulativen Dissertation befasst sich mit dem Einbinden solcher stochastischer Informationen in die Tourenplanung von Autotransportern.

Insbesondere stellt sich bei der dynamischen Planung der Transporte die Frage, welche der bereitstehenden Fahrzeuge ausgeliefert und welche auf die folgenden Tage verschoben werden sollen. Inspiriert durch diese praktische Anwendung betrachtet der zweite Beitrag dieser Dissertation ein dynamisches, mehrperiodisches Vehicle-Routing-Problem mit strikten Restriktionen an die Ladungskapazität sowie die Zeitfenster der Kundenaufträge. Im Mittelpunkt steht dabei die Untersuchung des Worst-Case-Verahltens verschiedener Online-Algorithmen.

In der Regel wird der Transport der Autos nicht vom Automobilunternehmen, dem Original Equipment Manufacturer (OEM), selbst durchgeführt, sondern an spezialisierte Logistikdienstleister ausgelagert. Diese Tendenz zum Outsourcing von logistischen Aufgaben ist bei weitem nicht auf die Distribution beschränkt und gewinnt stetig an Bedeutung (vgl. Klug, 2010, S. 125f.). Voss (2007, S. 225 f.) unterscheidet zwischen drei Dienstleistertypen: den klassischen Dienstleistern, den Kontraktlogistik-Dienstleistern (auch 3rd Party Logistics Provider (3PL)) und den koordinierenden Dienstleistern (auch 4th Party Logistics Provider (4PL)). Während sich erstere, zu denen beispielsweise Speditionen zählen, auf klassische Logistikdienstleistungen wie den Transport beschränken, bieten 3PLs ein breiteres und mittel- bis langfristig angelegtes Leistungsspektrum. Die 4PLs hingegen übernehmen üblicherweise lediglich steuernde Aufgaben in der Supply-Chain und besitzen selbst keine Anlagen.

Für den OEM stellt sich die grundsätzliche Frage, ob ein oder mehrere Logistikdienstleister für den Transport in Anspruch genommen werden sollen. Letzteres, welches analog zu bekannten Konzepten aus der Beschaffung als *multiple sourcing* bezeichnet werden kann, reduziert beispielsweise Abhängigkeiten von externen Dienstleistern (vgl. Schönsleben, 2011, S. 86). Außerdem sind Wechsel der Dienstleister verglichen mit einem *single sourcing* Ansatz deutlich leichter realisierbar, wodurch ein höherer Preisdruck ausgeübt werden kann (vgl. Werner, 2010, S. 150). Im Bereich des Supply-Chain-Managements gibt es zahlreiche Untersuchungen, die *single* und *multiple sourcing* miteinander vergleichen, siehe zum Beispiel Burke et al. (2007) sowie Berger und Zeng (2006). Bei Anwendung eines *multiple sourcing* Ansatzes ist eine übliche Vorgehensweise bei der Aufteilung der Auftragsumfänge auf die verschiedenen Anbieter die Festlegung von prozentualen Quoten. So wäre bei drei ausgewählten Anbietern denkbar, einem Partner 50% des Auftragsvolumens zuzuteilen und den anderen beiden je 25%. Bei der Verteilung von Transportrouten auf mehrere Dienstleister stellt sich allerdings die Frage anhand welcher Kenngröße die Aufträge verteilt werden sollen. Würde man beispielsweise lediglich darauf achten, die Zahl der Touren anhand der Quoten zu verteilen, könnte dies zu einem ungewollten Ungleichgewicht bei der Länge der Touren führen. Eine faire Aufteilung der Transportaufträge auf mehrere Anbieter anhand verschiedener Kriterien hat daher der dritte Beitrag innerhalb dieser Dissertation zum Ziel. Das dabei aufgestellte Modell ist jedoch generisch und somit auf weitere Problemstellungen anwendbar.

2 Kurzbeschreibung der Beiträge

Im folgenden Abschnitt werden die einzelnen Beiträge kurz zusammengefasst. Dabei wird insbesondere darauf eingegangen, inwiefern diese einen Beitrag zur Schließung von Forschungslücken leisten und welche zentralen Ergebnisse sie liefern.

Beitrag B1: A multiperiod auto-carrier transportation problem with probabilistic future demands

In diesem Beitrag wird untersucht, wie die Auslieferung fertiger Autos ausgehend von einem zentralen Terminal (Depot) an verschiedene Händler optimiert werden kann. Dazu wird ein mehrperiodisches Problem betrachtet, bei dem täglich über die auszuliefernden Autos und die Routen der Autotransporter entschieden werden muss. Im Mittelpunkt der Forschungsarbeit steht dabei die bisher in der Literatur vernachlässigte Frage, ob die Antizipation zukünftiger Händleraufträge eine Kostenersparnis für die Tourenplanung generieren kann. In der Modellierung wird dafür angenommen, dass Händler-spezifische Wahrscheinlichkeiten für eingehende Aufträge vorliegen, die beispielsweise aus historischen Daten geschätzt werden können. Die betrachtete Problemstellung gehört somit in die Klasse der stochastischen, dynamischen Tourenplanungsprobleme, wobei zusätzlich spezielle Kapazitätsrestriktionen der Autotransporter berücksichtigt werden müssen. Diese werden über *loading patterns* modelliert, welche von Hu et al. (2015) eingeführt wurden. Wie bereits im vorigen Abschnitt aufgezeigt wurde, ist bei der Fahrzeugdistribution neben geringen Kosten - in diesem Beitrag zusammengesetzt aus Kosten pro Tour, pro Händlerstopp und pro gefahrenen Kilometer - auf den Aspekt der Liefertreue zu achten. Dieser wird durch auftragsabhängige *due dates* berücksichtigt, welche nicht überschritten werden dürfen.

Zur Lösung der Problemstellung wird eine dreistufige Heuristik präsentiert, welche jeweils zu Beginn einer Periode angewandt wird, um die Auslieferungstouren des Tages festzulegen. Im ersten Schritt findet die Auswahl der auszuliefernden Autos statt, wobei die stochastische Komponente Berücksichtigung findet. Nach der Tourenbildung im zweiten Schritt werden gegebenenfalls auftretende Leerkapaziäten genutzt und nicht-dringende Touren aufgeschoben. Dieser Lösungsansatz wird mit grundlegenden Strategien im Rahmen einer Fallstudie verglichen, welche auf praxisnahen Daten von Wensing (2018) beruht. Dabei zeigt sich, dass sich durch die Antizipation signifikante Verbesserungen ergeben. Außerdem lässt sich feststellen, dass der Effekt umso größer ist, je weniger Aufträge eingehen, wodurch die Kundendichte pro Tag abnimmt.

Beitrag B2: The multiperiod 2-customer vehicle routing problem

Bedingt durch die Komplexität der praktischen Problemstellung muss in Beitrag B1 zur Lösung auf heuristische Verfahren zurückgegriffen werden. Um ein besseres Verständnis für die Schwierigkeit der Auswahl von auszuführenden Aufträgen in dynamischen Tourenplanungsproblemen zu erlangen, wird in diesem Beitrag ein dynamisches VRP mit engen Restriktionen betrachtet. Alle Kunden haben einen einheitlichen Bedarf und auf jeder Tour können maximal zwei Kunden bedient werden. Abgesehen von Kunden, die in der ersten oder letzten Periode bedient werden müssen, stehen zwei aufeinanderfolgende Zeitperioden zur Auslieferung zur Verfügung, wobei die Aufträge erst zu Beginn der ersten möglichen Auslieferungsperiode bekannt werden. Sobald über die Auswahl an Kunden in einer Periode entschieden wurde, können die kürzestmöglichen Touren aufgrund der Limitierung auf zwei Kunden pro Tour exakt in polynomieller Zeit mittels des Algorithmus von Edmonds

(1965) bestimmt werden, indem das VRP in ein Matching-Problem überführt wird. Im Zentrum des Beitrags steht die Frage, wie gut die optimale Lösung unter vollständigem Wissen aller Kundenaufträge durch Algorithmen mit geschickter Kundenauswahl approximiert werden kann. Für die Analyse solcher Online-Algorithmen hat sich die Untersuchung der *competitive ratio* etabliert, welche das Verhalten im *worst case* betrachtet (vgl. Sleator und Tarjan (1985) sowie Karlin et al. (1988)). Diese steht auch im Mittelpunkt der Veröffentlichung von Angelelli et al. (2007), die den Fall eines unkapazitierten Fahrzeugs betrachten.

Es zeigt sich, dass selbst bei einem Planungszeitraum von lediglich zwei Perioden jeder Online-Algorithmus eine Lösung liefern kann, welche die Gesamtlänge der optimalen Lösung um den Faktor $\sqrt{2}$ übersteigt. Außerdem wird ein Algorithmus vorgestellt, der zumindest für eine Teilmenge an Instanzen diese *competitive ratio* von $\sqrt{2}$ garantiert. Numerische Tests deuten darauf hin, dass dieser Algorithmus den Faktor auch für beliebige Instanzen einhält. Für einen weiteren Algorithmus, den SMART(p)-Algorithmus, kann eine *competitive ratio* von 1,5 nachgewiesen werden. Diese Resultate können jedoch nicht auf mehr als zwei Perioden übertragen werden. Bereits für drei Perioden überschreitet die *competitive ratio* des SMART(p)-Algorithmus den Wert von 1,5 und es kann bewiesen werden, dass kein Algorithmus eine *competitive ratio* von $\sqrt{2}$ haben kann. Insbesondere die unteren Schranken für die *competitive ratio* liefern einen interessanten Einblick in die Güte von Online-Algorithmen im Bereich der dynamischen Tourenplanung, da sie auf komplexere Varianten der Problemstellung übertragen werden können.

Beitrag B3: Fair Task Allocation Problem

Wie bereits im einleitenden Kapitel festgestellt wurde, wird die Auslieferung von produzierten Autos an Händler oft von mehreren Logistikdienstleistern übernommen. Falls das Auftragsvolumen dabei vertraglich durch prozentuale Quoten fixiert ist, muss auf eine Verteilung der Aufträge geachtet werden, welche diese Quoten möglichst genau einhält. Motiviert durch diese Anwendung wird in diesem Beitrag ein generisches gemischt-ganzzahliges Programm aufgestellt. Dieses hat zum Ziel, die gewichteten summierten Abweichungen von den Zielvorgaben in verschiedenen Dimensionen, welche in der genannten praktischen Anwendung beispielsweise *Anzahl an Touren* und *Länge der Touren* sein können, zu minimieren. Optimierte Zuordnungen erhöhen dabei die Zufriedenheit der Dienstleister, bringen aber meist keinen unmittelbar messbaren monetären Vorteil. Daher ist das Ziel dieses Beitrags die Entwicklung einer effizienten Heuristik, die gute Lösungen für dieses Fairness-Problem erzeugt.

Dafür wird mit der Tabu-Suche auf eine Meta-Heuristik zurückgegriffen, welche sowohl auf realen Daten als auch auf generierten Instanzen getestet wird. Dabei zeigt sich, dass alle realen Instanzen exakt gelöst werden können und die Tabu-Suche auf den größeren, generierten Instanzen in über 85% der Fälle mindestens so gut ist wie die Lösung durch IBM ILOG CPLEX bei einem Zeitlimit von einer Stunde. Abgesehen davon werden in diesem Beitrag Komplexitätsaussagen zum betrachteten Problem getroffen, wobei sich zeigt, dass es sich um ein NP-schweres Problem im strengen Sinne handelt, das jedoch bei beschränkter Anzahl an Anbietern und Dimensionen exakt in pseudopolynomieller Zeit gelöst werden kann. Außerdem werden Zusammenhänge zu bekannten Fragestellungen aus dem Scheduling herausgearbeitet.

Literaturverzeichnis

Angelelli, E., Speranza, M. G., und Savelsbergh, M. W. P. (2007). Competitive analysis for dynamic multiperiod uncapacitated routing problems. *Networks*, 49(4):308–317.

Berger, P. und Zeng, A. (2006). Single versus multiple sourcing in the presence of risks. *The Journal of the Operational Research Society*, 57(3):250–261.

Bundesverband CarSharing (o.A.). Car-Sharing Entwicklung in Deutschland. https://carsharing.de/presse/pressemitteilungen/carsharing-jahresbilanz-2016-mehr-17-millionen-carsharing-nutzer. abgerufen am 26.02.2018.

Burke, G. J., Carrillo, J. E., und Vakharia, A. J. (2007). Single versus multiple supplier sourcing strategies. *European Journal of Operational Research*, 182(1):95–112.

Cats, O., Susilo, Y. O., und Reimal, T. (2017). The prospects of fare-free public transport: evidence from Tallinn. *Transportation*, 44(5):1083–1104.

Cordeau, J.-F., Dell'Amico, M., Falavigna, S., und Iori, M. (2015). A rolling horizon algorithm for auto-carrier transportation. *Transportation Research Part B*, 76:68–80.

Dannenberg, J. (2005). Von der Technik zum Kunden. In Gottschalk, B., Kalmbach, R. G., und Dannenberg, J. (Hrsg.), *Markenmanagement in der Automobilindustrie*. Gabler Verlag.

Dantzig, G. B. und Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.

Dell'Amico, M., Falavigna, S., und Iori, M. (2014). Optimization of a real-world auto-carrier transportation problem. *Transportation Science*, 49(2):402–419.

Edmonds, J. (1965). Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards*, 69(1 and 2):125–130.

Fandel, G., Giese, A., und Raubenheimer, H. (2009). *Supply Chain Management: Strategien - Planungsansätze - Controlling*. Springer, Berlin.

Grafmüller, M. H. (2000). *Prozeßmanagement in der Automobilindustrie: Betriebliche Umsetzung am Beispiel der Volkswagen AG*. Gabler Verlag, Wiesbaden.

Herold, L. (2005). *Kundenorientierte Prozesssteuerung in der Automobilindustrie*. DUV/GWV Fachverlage GmbH, Wiesbaden.

Hu, Z.-H., Zhao, Y., Tao, S., und Sheng, Z.-H. (2015). Finished-vehicle transporter routing problem solved by loading pattern discovery. *Annals of Operations Research*, 234(1):37–56.

Huber, A. und Laverentz, K. (2012). *Logistik*. Vahlen, München.

Ihme, J. (2006). *Logistik im Automobilbau*. Hanser, München.

Karlin, A. R., Manasse, M. S., Rudolph, L., und Sleator, D. D. (1988). Competitive snoopy caching. *Algorithmica*, 3(1-4):79–119.

Klug, F. (2010). *Logistikmanagement in der Automobilindustrie*. Springer, Heidelberg.

McKinsey&Company (2016). Autoindustrie: Langfristiger Wachstumskurs ist intakt - aber Erlösquellen verändern sich. https://www.mckinsey.de/sites/mck_files/files/160104_pm_auto2030.pdf. abgerufen am 26.02.2018.

OICA (2018a). Production statistics. http://www.oica.net/category/production-statistics. abgerufen am 26.02.2018.

OICA (2018b). World vehicles in use - all vehicles. http://www.oica.net/category/vehicles-in-use. abgerufen am 26.02.2018.

Ritzinger, U., Puchinger, J., und Hartl, R. F. (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231.

Schönsleben, P. (2011). *Integrales Logistikmanagement: Operations und Supply Chain Management innerhalb des Unternehmens und unternehmensübergreifend*. Springer, Heidelberg, 6. Auflage.

Skala-Gast, D. (2012). Zusammenhang zwischen Kundenzufriedenheit und Kundenloyalität: Eine empirische Analyse am Beispiel der deutschen Automobilindustrie. In Burmann, C. und Kirchgeorg, M. (Hrsg.), *Innovatives Markenmanagement*, Band 38. Springer Gabler, Wiesbaden.

Sleator, D. D. und Tarjan, R. E. (1985). Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208.

Stautner, U. (2001). *Kundenorientierte Lagerfertigung im Automobilvertrieb*. DUV GmbH, Wiesbaden.

Tadei, R., Perboli, G., und Della Croce, F. (2002). A heuristic algorithm for the auto-carrier transportation problem. *Transportation Science*, 36(1):55–62.

Toth, P. und Vigo, D. (Hrsg.) (2002). *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia.

Voss, H. (2007). Supra-Adaptivität durch Outsourcing - Dienstleistertypen und Kompetenzprofile. In Günthner, W. A. (Hrsg.), *Neue Wege in der Automobillogistik - Die Vision der Supra-Adaptivität*, Seiten 219–230. Springer, Berlin.

Wensing, T. (2018). Social and environmental impact of advances in economically-driven transport optimization: Case study in automobile distribution. In Brandenburg, M., Hahn, G. J., und Rebs, T. (Hrsg.), *Social and Environmental Dimensions of Organizations and Supply Chains*, Seiten 195–210. Springer.

Werner, H. (2010). *Supply Chain Management: Grundlagen, Strategien, Instrumente und Controlling*. Gabler, Wiesbaden, 4. Auflage.

Werthmann, D., Ruthenbeck, C., und Scholz-Reiter, B. (2012). Standardisierter Informationsaustausch in Wertschöpfungsnetzwerken der Automobilindustrie. In Schenk, M., Zadek, H., Müller, G., Richter, K., und Seidel, H. (Hrsg.), *Sichere und nachhaltige Logistik*, Seiten 145–151.

II Beitrag B1

A multiperiod auto-carrier transportation problem with probabilistic future demands

Billing, C., Jaehn, F., and Wensing, T. (2018). A multiperiod auto-carrier transportation problem with probabilistic future demands. *Journal of Business Economics*, 88:1009-1028.
https://doi.org/10.1007/s11573-018-0904-x

Abstract

In this paper we study the problem of delivering finished vehicles from a logistics yard to dealer locations at which they are sold. The requests for cars arrive dynamically and are not announced in advance to the logistics provider who is granted a certain time-span until which a delivery has to be fulfilled. In a real-world setting, the underlying network is relatively stable in time, since it is usually a rare event that a new dealership opens or an existing one terminates its service. Therefore, probabilities for incoming requests can be derived from historical data. The study explores the potential of using such probabilities to improve the day-to-day decision of sending out or postponing cars that are ready for delivery. Apart from the order selection, we elaborate a heuristic to optimize delivery routes for the selected vehicles, whereby special loading constraints are considered to meet the particular constraints of car transportation via road. In a case study, we illustrate the value of introducing probabilistic information to the planning process and compare the quality of different configurations of our approach.

III Beitrag B2

The multiperiod 2-customer vehicle routing problem

Christian Billing

University of Augsburg

Abstract

Few combinatorial optimization problems have attracted as much attention as vehicle routing problems. Most of publications deal with very complex variants of the problem, so performance guarantees of solution approaches are hard to derive. In contrast, in this work we consider a dynamic multi-period vehicle routing problem with restrictive conditions. We assume that no more than two customers can be visited on a tour. The orders of customers arrive dynamically at the beginning of the periods and must be fulfilled until a given deadline. As the problem in a one-period setting is easy to solve by matching algorithms, the difficulty is to decide about the service-period of the customers. It is shown that no algortihm has a competitive ratio of less than $\sqrt{2}$ in a two-period setting and this lower bound even increases for a larger planning horizon. Furthermore, tight competitive ratios of different solution approaches are verified.

1 Introduction

Vehicle routing problems belong to the most studied topics among combinatorial optimization problems. All variants of the problem have in common the search of a cost-minimal set of routes to serve a given set of customers by a fleet of vehicles that usually have limited capacity. Beyond that, a wide range of problem variations exists, such as the existence of multiple depots, pick-up and delivery customers, or time-windows. An extensive survey on the field of vehicle routing problems is given by Toth and Vigo (2002). As in many real-world problems some of the relevant data is not known in advance, the research strand of dynamic and stochastic routing problems is of great interest. A recent survey is given by Ritzinger et al. (2016). Most of the research focusses on the development of heuristics to derive high quality solutions, often motivated by practical applications. However, literature about the competitiveness of algorithms is rather scarce, due to the complexity of the problems. According to Larsen et al. (2008), only very simplified versions of dynamic routing problems can be investigated by competitive analysis. This paper explores the competitiveness of solution approaches for a dynamic vehicle routing problem with strict restrictions on the capacity of the vehicles as well as on the demand and the time windows of customers.

An interesting variant of the vehicle routing problem is the case of an unitary demand of one and a vehicle capacity of k, so the number of customers on every route is limited to k. This problem is known as *k-customer vehicle routing problem*, see for example Hassin and Rubinstein (2005), or as *k-tour cover problem*, see for example Asano et al. (1997). Minimizing the covered distance is NP-hard for $k \geq 3$ as shown by Archetti et al. (2005) even if the distances between customers satisfy the triangle inequality. For the metric case of the k-customer VRP, Haimovich et al. (1988) show that the *iterated optimal tour partition*, introduced in Haimovich and Rinnooy Kan (1985), has an approximation factor of $2 - \frac{1}{k}$. This bound is tight as shown by Li and Simchi-Levi (1990). If the solution of the travelling salesman problem, which is part of the heuristic, is determined by an α-approximation algorithm instead of an exact approach, the approximation factor increases to $1 + (1 - \frac{1}{k}) \cdot \alpha$. For example, using the algorithm of Christofides (1976) with $\alpha = 1.5$, the approximation factor is 2 for $k = 3$. This factor was slightly improved by Bazgan et al. (2005), who proved that the metric 3-customer VRP is $\frac{197}{99} + \epsilon$ standard approximable for every $\epsilon > 0$.

Hassin and Rubinstein (2005) state that the 2-customer vehicle routing problem with arbitrary non-negative costs can be transformed into a minimum weighted matching problem. This can be solved in polynomial time by the algorithm of Edmonds (1965). Archetti et al. (2005) analyse the skip delivery problem, which can be applied to practical problems involving the transportation of big containers, e.g. for waste collection. In their problem formulation the demands are allowed to be greater than one, while the capacity is still two. They prove that this problem remains solvable in polynomial time by transformation to a minimum weighted b-matching problem. In the case of symmetrical costs satisfying the triangle inequality, they show that an optimal solution exists in which every customer with demand greater than two is visited by as many direct, fully loaded trips as possible. Hence, the skip delivery problem reduces to the 2-customer vehicle routing problem if the costs are a metric.

In this paper, we investigate an extension of the 2-customer vehicle routing problem to T periods. The service requests of customers arrive at the beginning of a period and may be fulfilled at the same or the next period. So, the crucial decision is about serving or postponing the incoming orders. Once this decision has been made, the daily routing problem can be solved exactly by the algorithm of Edmonds (1965). The objective is to serve all customers by routes of minimum total length.

The decision about serving or postponing customers is important in a wide range of dynamic vehicle routing problems. For example, Wen et al. (2010) investigate a routing problem over multiple periods with dynamically arriving service requests that must be fulfilled until a given deadline. The consideration of probabilities for future requests is included in Albareda-Sambola et al. (2014). Obviously, the possibility to postpone customers to subsequent periods is pertinent to many practical problems. An application in the field of courier services is considerd by Angelelli et al. (2009). The delivery of finished vehicles by auto-carriers in a dynamic setting is investigated in Billing et al. (2018).

To the best of my knowledge, there is no existing literature about k-customer vehicle routing problems over multiple periods. However, a very similar problem has been investigated by Angelelli et al. (2007). In contrast to the problem considered here, they assume a single vehicle with unlimited capacity. Hence, after the decision about the set of customers to be served, a travelling salesman problem must be solved. Different solution strategies are presented that are investigated according to their competitive ratio when customers are located on the non-negative real-line or on the Euclidean plane. Furthermore, lower bounds for the competitive ratio of any online algorithm are presented, both for the case of $T = 2$ and $T \geq 3$. Some of the results presented here are based on findings of Angelelli et al. (2007).

The remainder of the paper is organized as follows. In Section 2 we give a formal problem description and introduce the notation. Afterwards, we investigate the offline problem when all orders are known in advance. In Section 4, lower bounds for the competitive ratio of online algorithms are presented as well as an analysis of some solution strategies for the case of two periods. The extension to $T \geq 3$ is investigated in Section 5, while Section 6 concludes the study.

2 Problem definition and notation

We regard a vehicle routing problem over multiple periods with a single depot. An arbitrarily large fleet of vehicles with a capacity of two is given. We assume that the distances between all customers (including the depot) are non-negative and satisfy the triangle inequality. All customers have a demand of one, which implies that at most two customers are allowed per tour. The customers have to be served in one of two given consecutive feasible periods. Additionally, there are customers that must be served in the first period and customers that must be served in the last period. The focus of this study will be on the online problem, when customer requests are not known in advance but arise at the beginning of its first feasible service period. In every period we have to decide about the set of customers to be served and about the routes that must start and end at the depot. The objective is to minimize the sum of distances over the routes of all periods.

The notation is based on Angelelli et al. (2007), who consider the uncapacitated version of this problem. The planning horizon is divided into periods $1, 2, ..., T$. By C_1 and C_T we denote the set of customers that must be served in the first and in the last period, respectively. Customers that can be served in periods t or $t+1$ are denoted by $C_{t|t+1}$. The set of all customers is given by $C = C_1 \cup C_{1|2} \cup \ldots \cup C_{T-1|T} \cup C_T$. The distance between two customers v_i and v_j is denoted by $d(v_i, v_j)$. Accordingly, the distance between the depot D and a customer v_i is given by $d(D, v_i)$. As the distances form a metric, they are symmetrically and satisfy the triangle inequality. Primarily, we will focus on the online version of the problem when customer sets $C_{t|t+1}$ and C_T are not known before period t and T, respectively. So, the decision about the routes in period $t < T$ must be made without knowledge about customers $C_{t+1|t+2}, ..., C_{T-1|T}$ and C_T. In the first period, all customers in C_1 must be served and we have to decide about the subset of $C_{1|2}$ to serve. In general, in period t all orders of customers in $C_{t-1|t}$ not fulfilled in $t-1$ must be fulfilled and we have to decide about the subset of $C_{t|t+1}$ to serve. In period T, the set of customers to visit is predefined by the decision in $T-1$.

A popular way to measure the quality of an algorithm for online problems is to compare its solution to an optimal solution of the problem with complete knowledge. The idea was introduced by Sleator and Tarjan (1985) and later formalized by Karlin et al. (1988). The performance of an online algorithm A in the worst case can be described by the competitive ratio

$$CR_A = \max_I \frac{z^A(I)}{z^*(I)} \geq 1, \tag{1}$$

where $z^*(I)$ denotes the value of an optimal solution of instance I with all information given a priori and $z^A(I)$ denotes the solution value of algorithm A for instance I.

3 Offline problem

First of all, we have a look at the offline version of the problem, when all customer sets (C_1, C_T and $C_{t|t+1}$ $\forall t = 1, 2, ..., T-1$) are known a priori. So, we have complete knowledge and all routes can be planned right at the beginning. As every vehicle visits at most two customers due to the capacity constraint, every feasible solution contains the way between the depot D and customer u at least once for every customer $u \in C$. Hence, these distances cannot be avoided and we derive a lower bound for the solution value:

$$z^*(I) \geq \sum_{u \in C} d(D, u). \tag{2}$$

In addition to this unavoidable length, further distances have to be covered in general, which we will call *matching distances*. Firstly, when customers u and v are served together in the solution, the distance $d(u, v)$ is travelled. Secondly, if customer u is the only customer on a tour, we call it an *oscillation tour* with an additional distance of $d(D, u)$.

As the triangle inequality holds for the distances, the worst solution is serving every customer by an oscillation tour. This causes a matching distance of $\sum_{u \in C} d(D, u)$

and a total length of $2 \cdot \sum_{u \in C} d(D, u)$. Due to the lower bound (2) we can state the following.

Theorem 1. *The total length of every feasible solution of the problem is at most twice the total length of the optimal solution.*

Although oscillation tours seem to be unfavourable, the optimal solution can have more oscillation tours than necessary as we can see in the following example.

Example. We consider two time periods with customers $C_1 = \{v_1\}, C_{1|2} = \{v_2, v_3\}$ and $C_2 = \{v_4\}$ that are located in the plane as depicted in Figure 1.

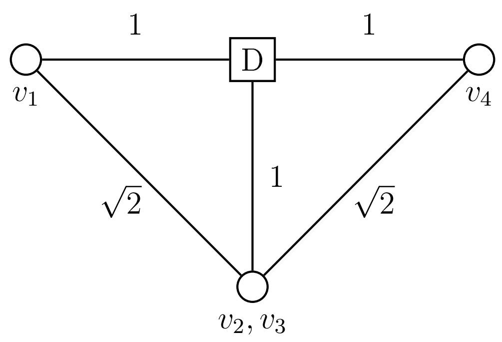

Figure 1: Example of profitable oscillation tours

The best solution consists of three tours $(D, v_1, D), (D, v_2, v_3, D), (D, v_4, D)$ with a total length of 6. The solution $(D, v_1, v_2, D), (D, v_3, v_4, D)$ without oscillation tours has length $4 + 2\sqrt{2} > 6$.

Nonetheless, we can establish an upper bound on the number of oscillation tours in an optimal solution. This upper bound can be used to accelerate the exact solution procedure presented afterwards.

Lemma 1. *There is an optimal solution such that two customers are not served by oscillation tours if they could be served in the same period.*

This lemma, which is a consequence of the triangle inequality, implies that there is an optimal solution with at most one oscillation tour per period. So, the total number of oscillation tours is bounded by T. The following theorem shows that the number of oscillation tours can be reduced even further.

Theorem 2. *There is an optimal solution without an oscillation tour in even periods, possibly except for period T. Thus, the total number of oscillation tours in an optimal solution is at most $\lceil \frac{T-1}{2} \rceil + 1$.*

Proof: We consider an optimal solution satisfying Lemma 1. We assume that the solution has an oscillation tour for a customer u in period $t \neq T$ with t even. As $u \in C_{t-1|t} \vee u \in C_{t|t+1}$ the oscillation tour can be executed in the odd period $t-1$ or $t+1$, respectively. By this we can shift every oscillation tour to an odd period. As our solution has the property of Lemma 1, there is no other oscillation tour in this period. Hence, we can avoid oscillation tours in even periods, except for period T and the total number of oscillation tours is bounded by $\lceil \frac{T-1}{2} \rceil + 1$. □

To solve the problem exactly, we transform it into a minimum weighted perfect matching problem that can be solved exactly in polynomial time by the blossom-algorithm of Edmonds (1965). As mentioned before, distances $\sum_{u \in C} d(D,u)$ cannot be avoided, so our objective is to minimize the additional matching distances. To obtain an instance of the minimum weighted perfect matching problem, every customer is represented by a *customer node* and two customer nodes are connected by an edge if the corresponding customers have a common feasible service period. The weight of an edge between u and v is given by $d(u,v)$. Additionally, we use *oscillation nodes* o_t to enable oscillation tours in the solution. An oscillation node o_t is connected with a customer node u if u can be served in period t and the weight of the edge is given by $d(D,u)$. Because of Theorem 2, oscillation nodes are necessary only for periods T and $t < T$ with t odd. Besides, all oscillation nodes are connected among each other by an edge with weight 0. If the total number of nodes is odd, one additional dummy node v_D is necessary to enable a perfect matching. The dummy node v_D is connected to all oscillation nodes with a weight of 0. The construction of the matching instance is illustrated for $T = 4$ in Figure 2.

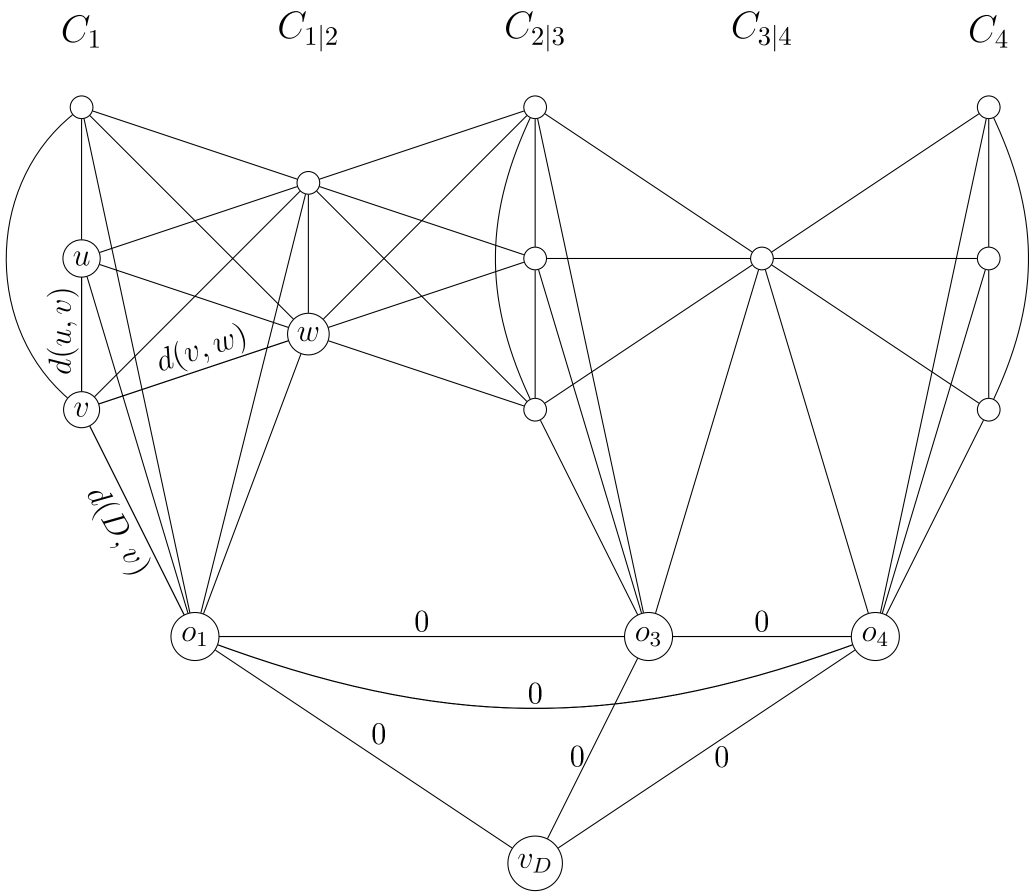

Figure 2: Construction of the matching instance for the solution of the offline problem

We obtain the optimal solution of the routing problem as follows. Two customers u and v are served together if the edge (u, v) between the customer nodes u and v is part of the minimum weighted perfect matching. Furthermore, an oscillation tour is executed for customer u in period t if the minimum weighted perfect matching contains an edge (u, o_t).

4 Results for the online problem with two periods

From now on, we consider the online problem, i.e. orders of customers in $C_{t|t+1}$ arrive at the beginning of period t and the decision about the subset of $C_{t|t+1}$ to be served in period t must be made without any knowledge about the future. The construction of the optimal routes for the selected customers in every period can be done exactly by the algorithm of Edmonds (1965). In this section we investigate the case of $T = 2$, where the crucial decision is about the subset of customers in $C_{1|2}$ to be served in the first period.

Basic strategies and lower bounds

We start with the two basic strategies IMMEDIATE - all customers of $C_{1|2}$ are served in period one - and DELAY - all customers of $C_{1|2}$ are served in period two, which were introduced for the uncapacitated version by Angelelli et al. (2007). However, these strategies do not lead to good solutions in the worst case:

Theorem 3. *The strategies IMMEDIATE and DELAY both have competitive ratio 2.*

Proof: Due to Theorem 1, every algorithm A provides $CR_A \leq 2$. Thus, it is sufficient to show $CR_{\text{IMMEDIATE}} \geq 2$. Therefore consider an instance with $C_1 = \emptyset$, $C_{1|2} = \{v\}$ and $C_2 = \{u\}$, where v and u have the same location $(d(u, v) = 0 \wedge d(D, u) = d(D, v))$. The optimal solution serves both customers in a common route in period two with length $2 \cdot d(D, v)$, while IMMEDIATE leads to two single routes of total length $4 \cdot d(D, v)$. For the strategy DELAY an analogous example with customers in C_1 and $C_{1|2}$ verifies the statement. $\square$

Next, we want to analyse the class of algorithms that decide to serve all customers of $C_{1|2}$ in the same period. These algorithms apply IMMEDIATE or DELAY depending on the customer sets C_1 and $C_{1|2}$. By using the information about customers C_1

and $C_{1|2}$ the competitive ratio can be improved as we will see later. However, there is a lower bound for the competitive ratio of such algorithms.

Theorem 4. *Let A be an online algorithm that decides to serve all customers of $C_{1|2}$ in the same period. Then $CR_A \geq \frac{3}{2}$.*

Proof: Consider the following situation with customers $C_1 = \{v_1\}$ and $C_{1|2} = \{v_2, v_3\}$ located in the plane as shown in Figure 3

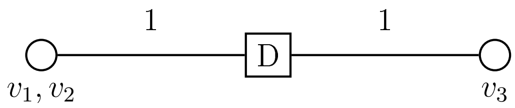

Figure 3: Location of customers in the plane

Case 1: Algorithm A decides to serve customers v_2 and v_3 in $t = 1$.

Let $C_2 = \{v_4\}$ and v_4 has the same position as v_3. Then, the optimal solution - serving v_1 and v_2 in a common route in $t = 1$ and v_3 and v_4 together in $t = 2$ - has length 4. However, the solution of algorithm A has length $6 = 3/2 \cdot 4$.

Case 2: Algorithm A decides to serve customers v_2 and v_3 in $t = 2$.

Let $C_2 = \emptyset$. Then, the optimal solution - serving v_1 and v_2 in a common route in $t = 1$ and v_3 for its own in $t = 1$ or $t = 2$ - has length 4. However, the solution of algorithm A again has length $6 = 3/2 \cdot 4$. □

Note that this lower bound also holds for Euclidean distances as the customers are located in the Euclidean plane in the proof of Theorem 4.

Angelelli et al. (2007) introduce the algorithm SMART(p) for the problem with one vehicle of unlimited capacity. This algorithm applies IMMEDIATE if the distance of serving customers $C_1 \cup C_{1|2}$ is at most p times the distance needed to serve customers C_1. In the other case all customers of $C_{1|2}$ are postponed, i.e. DELAY is applied. When $p < 1$ ia chosen, DELAY is applied independent of the instance. Hence, the competitive ratio is equal to two. On the other side, the competitive ratio of the SMART(p) algorithm reaches the lower bound of Theorem 4 for $p = 2$.

Theorem 5. *$CR_{SMART(2)} = \frac{3}{2}$ for $T = 2$ periods.*

Proof: Theorem 4 implies $CR_{\text{SMART(2)}} \geq \frac{3}{2}$. For the evidence of $CR_{\text{SMART(2)}} \leq \frac{3}{2}$ we refer to the proof of theorem 9 in Angelelli et al. (2007) that remains correct for our problem with two customers per tour. □

Because of Theorem 4 and Theorem 5 we conclude that $p = 2$ is the best choice for p using the SMART algorithm for $T = 2$.

Corollary 1. *$CR_{SMART(2)} \leq CR_{SMART(p)}$ $\forall p \in \mathbb{R}$.*

Of course, it can be profitable to split the customers $C_{1|2}$, i.e. serving a subset in $t = 1$ and postponing the remaining customers to $t = 2$. However, we can verify the following lower bound for the competitive ratio of any algorithm.

Theorem 6. *Let A be an arbitrary online algorithm for $T = 2$. Then $CR_A \geq \sqrt{2}$.*

Proof: We can transfer the proof of Angelelli et al. (2007) for the same statement when one vehicle of unlimited capacity is available and customers are placed on the non-negative real line. Consider the situation below with $C_1 = \{v_1\}$ and $C_{1|2} = \{v_2\}$.

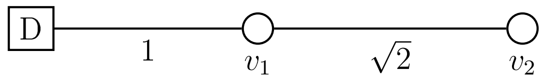

Figure 4: Location of customers in the plane in $t = 1$

Case 1: Algorithm A decides to serve customer v_2 in $t = 1$.

Let $C_2 = \{v_3\}$ with v_3 located at the same position as v_2. Then, algorithm A leads to routes (D, v_1, v_2, D) in $t = 1$ and (D, v_3, D) in $t = 2$ with a total length of

$4 \cdot (1+\sqrt{2})$. However, the optimal routes are (D, v_1, D) in $t = 1$ and (D, v_2, v_3, D) in $t = 2$ with a total length of $2 + 2 \cdot (1 + \sqrt{2})$. Hence, we have

$$CR_A \geq \frac{4 \cdot (1+\sqrt{2})}{2 + 2 \cdot (1+\sqrt{2})} = \sqrt{2}. \tag{3}$$

Case 2: Algorithm A decides to postpone customer v_2 to $t = 2$.

Let $C_2 = \emptyset$. Then, algorithm A leads to routes (D, v_1, D) in $t = 1$ and (D, v_2, D) in $t = 2$ with a total length of $2 + 2 \cdot (1 + \sqrt{2})$. However, the optimal solution is to serve both customers in a common route (D, v_1, v_2, D) in $t = 1$ with a total length of $2 \cdot (1 + \sqrt{2})$. Hence, we have

$$CR_A \geq \frac{2 + 2 \cdot (1+\sqrt{2})}{2 \cdot (1+\sqrt{2})} = \sqrt{2}. \tag{4}$$

Thus, the solution of algorithm A can be $\sqrt{2}$ times as long as the optimal solution, independent of the decision in $t = 1$. □

Match(λ)-algorithm

The following algorithm is developed in order to satisfy this lower bound of $\sqrt{2}$. The idea of the algorithm is to solve a minimum weighted matching problem in an auxiliary graph G_1 in the first period. The solution of the matching problem provides the subset $C^p_{1|2}$ of customers postponed to period two as well as routes for the customers $C_1 \cup C_{1|2} \setminus C^p_{1|2}$ served in period one. In period two the set of customers that must be served is predetermined and the optimal routes can be identified by a minimum weighted matching in an auxiliary graph G_2.

The graph G_1 contains a node u for every customer $u \in C_1 \cup C_{1|2}$. Additionally, one node o_1 is added that represents the possibility of an oscillation tour in period one. As customers $C_{1|2}$ can be postponed, a *postponing node* p_u is inserted for every customer $u \in C_{1|2}$. The set of all postponing nodes is denoted by $P_{1|2}$. If the number of nodes is odd, which is the case if $|C_1|$ is even, a dummy node d is added to enable a perfect matching. Hence, the set of nodes V_1 of graph G_1 is given by

$$V_1 = \begin{cases} C_1 \cup C_{1|2} \cup \{o_1\} \cup P_{1|2} & \text{if } |C_1| \equiv 1 \pmod 2 \\ C_1 \cup C_{1|2} \cup \{o_1\} \cup P_{1|2} \cup \{d\} & \text{if } |C_1| \equiv 0 \pmod 2 \end{cases}$$

The weights of the edges are defined as follows:

$$c(u,v) = \begin{cases} d(u,v) & \text{if } u \in C_1 \text{ and } v \in C_1 \cup C_{1|2} \\ d(u,D) & \text{if } u \in C_1 \text{ and } v = o_1 \\ \lambda \cdot d(u,D) & \text{if } u \in C_{1|2} \text{ and } v = p_u \\ 0 & \text{if } u,v \in \{o_1, d\} \cup P_{1|2} \\ \infty & \text{else} \end{cases}$$

All edges with at least one node in C_1 are weighted according to the corresponding matching distances. Consequently, edges (o_1, u) with $u \in C_1$ are weighted by the distance between customer u and the depot D and edges (u,v) with $u \in C_1$ and $v \in C_1 \cup C_{1|2}$ are weighted by the distance between the customers. Please note that there is no edge between customers within $C_{1|2}$ as it is never preferable to serve them on a common route in $t = 1$ compared to postponing both customers to the second period. The weights of edges (u, p_u) can be regarded as estimated costs for postponing customer u to the second period. The lower the value of λ, the more likely customers in $C_{1|2}$ are postponed to the second period. The construction of G_1 is illustrated in Figure 5, where edges of weight ∞ are disregarded.

In G_1, the minimum weighted perfect matching M_1 is determined. For edges $(u,v) \in M_1$ with $u,v \in C_1 \cup C_{1|2}$ the tour (D,u,v,D) is executed. If (u, o_1) with $u \in C_1$ is an element of the matching, we serve customer u by an oscillation tour. As the

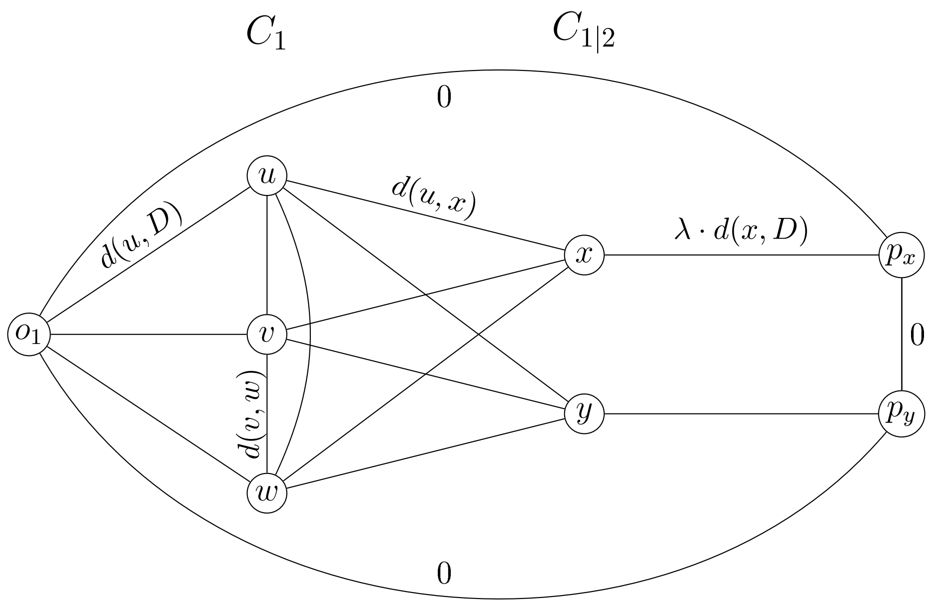

Figure 5: Construction of graph G_1

weights of these edges correspond to the matching distances, these tours are best for serving this subset of C. Customers $u \in C_{1|2}$ matched with its postponing node p_u are collected in $C^p_{1|2}$ and will be served in $t = 2$.

To find the best routes in the second period we construct the graph G_2 with nodes for every customer in $C^p_{1|2} \cup C_2$. The weight of the edges between these nodes is given by the distance between the corresponding customers. If the number of nodes is odd a node o_2 is inserted connected to all other nodes. The weight of these edges is given by the distance between the corresponding customer and the depot. The minimum weighted perfect matching M_2 in G_2 provides the best routes in $t = 2$. Again, customers are served together if their nodes are matched and a customer matched with o_2 is served by an oscillation tour. The algorithm is summarized in the following pseudocode.

Conjecture 1. *The competitive ratio of the Match(λ)-algorithm is minimal for $\lambda = 3 - 2\sqrt{2}$ with a competitive ratio of $\sqrt{2}$.*

Algorithm 1: Match(λ)-algorithm

Determine the minimum weighted perfect matching M_1 in G_1
for $(u, v) \in M_1$ **do**
  if $(u, v \in C_1 \cup C_{1|2})$ **then**
    execute tour (D, u, v, D) in $t = 1$
  else if $(u \in C_1 \wedge v = o_1)$ **then**
    execute tour (D, u, D) in $t = 1$
  else if $(u \in C_{1|2} \wedge v = p_u)$ **then**
    insert u into set $C^p_{1|2}$
end for
Determine minimum weighted perfect matching M_2 in G_2
for $(u, v) \in M_2$ **do**
  if $(u, v \in C^p_{1|2} \cup C_2)$ **then**
    execute tour (D, u, v, D) in $t = 2$
  end if
  else if $(u \in C^p_{1|2} \cup C_2 \wedge v = o_2)$ **then**
    execute tour (D, u, D) in $t = 2$
end for

Call in mind the instance of Figure 4 in the proof of Theorem 6. For $\lambda = 3-2\sqrt{2}$ the Match(λ)-algorithm is indifferent between serving customer v_2 in $t = 1$ or postponing it. If $\lambda > 3 - 2\sqrt{2}$, the Match(λ)-algorithm serves customer v_2 in the first period. This still holds when v_2 is shifted to the right by an arbitrarily small $\epsilon > 0$. However, if $C_2 = \{v_3\}$ with v_3 located at the same position as v_2, the ratio between the algorithmic solution value $z^{MA}(I)$ and the optimal solution value $z^*(I)$ is:

$$CR_{\mathrm{MA}(\lambda)} \geq \frac{z^{\mathrm{MA}(\lambda)}(I)}{z^*(I)} = \frac{4 \cdot (1 + \sqrt{2} + \epsilon)}{2 + 2 \cdot (1 + \sqrt{2} + \epsilon)} = \sqrt{2} + \frac{(2 - \sqrt{2})\epsilon}{2 + \sqrt{2} + \epsilon} > \sqrt{2}. \qquad (5)$$

In an analogous way it can be shown that the competitive ratio is greater than $\sqrt{2}$ for $\lambda < 3 - 2\sqrt{2}$ by shifting v_2 to the left by an arbitrarily small $\epsilon > 0$ and setting $C_2 = \emptyset$. Hence, $\lambda = 3 - 2\sqrt{2}$ is the only choice that may lead to the competitive ratio of $\sqrt{2}$.

For instances with only one customer in set $C_{1|2}$ we can prove Conjecture 1.

Lemma 2. *For instances I with $|C_{1|2}| \leq 1$ it holds*

$$\frac{z^{MA(\lambda)}(I)}{z^*(I)} \leq \sqrt{2},$$

for $\lambda = 3 - 2\sqrt{2}$ with $z^{MA(\lambda)}(I)$ denoting the solution value of the Match(λ)-algorithm.

Proof: As the statement is trivial for $C_{1|2} = \emptyset$, we suppose $C_{1|2} = \{u\}$ for the rest of the proof. We denote by L_1^A and L_2^A the length of the routes of the algorithmic solution in $t = 1$ and $t = 2$, respectively. Accordingly, L_1^* and L_2^* are the length of the routes in the optimal solution in $t = 1$ and $t = 2$. We consider two different cases.

Case 1: Customer u is served in period one by the algorithm, however, in the optimal solution u is postponed to period two.

First, we observe

$$L_1^A \leq L_1^* + d(D, u) + (3 - 2\sqrt{2}) \cdot d(D, u). \tag{6}$$

When u is served in $t = 1$ by the Match($3-2\sqrt{2}$)-algorithm, the distance $d(D, u)$ cannot be avoided. However, the additional matching distance in the first period - in comparison to the optimal solution - is at most $(3-2\sqrt{2}) \cdot d(D, u)$ as otherwise customer u would have been postponed to the second period by the algorithm. (Note that the algorithm determines the best routes for the selected customers.)

As $L_2^* \geq \max\{L_2^A, 2 \cdot d(D,u)\}$ holds, we obtain the following in conjunction with inequality (6):

$$\begin{aligned} \frac{L_1^A + L_2^A}{L_1^* + L_2^*} &\leq \frac{L_1^A + L_2^A}{L_1^A + (2\sqrt{2}-4)\cdot d(D,u) + \max\{L_2^A, 2\cdot d(D,u)\}} \\ &\leq \frac{2\cdot d(D,u) + L_2^A}{(2\sqrt{2}-2)\cdot d(D,u) + \max\{L_2^A, 2\cdot d(D,u)\}} \\ &\leq \frac{4\cdot d(D,u)}{2\cdot\sqrt{2}\cdot d(D,u)} = \sqrt{2}. \end{aligned} \tag{7}$$

The second inequality holds as the non-negative value $L_1^A - 2\cdot d(D,u)$ is subtracted from the numerator and the denominator of a fraction with value greater than one. The same argument holds for the third inequality when $\max\{L_2^A, 2\cdot d(D,u)\} - 2\cdot d(D,u)$ is subtracted.

Case 2: Customer u is postponed to period two by the algorithm, however, in the optimal solution u is served in period one.

A similar argumentation as for inequality (6) and the obvious result $L_1^* \geq 2 \cdot d(D,u)$ lead to

$$L_1^* \geq \max\{L_1^A + d(D,u) + (3-2\sqrt{2})\cdot d(D,u), 2\cdot d(D,u)\}. \tag{8}$$

Not serving customer u in the second period in the optimal solution saves at most $2\cdot d(D,u)$, so we conclude

$$L_2^* \geq L_2^A - 2\cdot d(D,u). \tag{9}$$

Inequalities (8) and (9) imply

$$\frac{L_1^A + L_2^A}{L_1^* + L_2^*} \leq \frac{L_1^A + L_2^A}{\max\{L_1^A + (4 - 2\sqrt{2}) \cdot d(D,u), 2 \cdot d(D,u)\} + (L_2^A - 2 \cdot d(D,u))} = \frac{L_1^A + L_2^A}{\max\{L_1^A + (2 - 2\sqrt{2}) \cdot d(D,u), 0\} + L_2^A}. \tag{10}$$

Subtracting the non-negative values $\max\{L_1^A + (2 - 2\sqrt{2}) \cdot d(D,u), 0\}$ and $L_2^A - 2 \cdot d(D,u)$ results in

$$\frac{L_1^A + L_2^A}{L_1^* + L_2^*} \leq \frac{(2\sqrt{2} - 2) \cdot d(D,u) + 2 \cdot d(D,u)}{2 \cdot d(D,u)} = \sqrt{2}. \tag{11}$$

This completes the proof. □

It remains an open question if this factor holds for arbitrary instances. To achieve an indication, a computational study was applied with eight different test cases. The cases differ in the number of customers in the sets $C_1, C_{1|2}$ and C_2. For every test case, 200,000 instances were generated with customers randomly distributed in the $[-1,1] \times [-1,1]$ grid and the depot located in the origin. Every instance was solved by the Match$(3-2\sqrt{2})$-algorithm as well as exactly when assuming total knowledge about the set C_2. The worst case ratio and the average ratio are shown in Table 1.

$\lvert C_1\rvert$	$\lvert C_{1\|2}\rvert$	$\lvert C_2\rvert$	Worst ratio	Average ratio
1	1	1	1.388	1.017
1	2	1	1.338	1.023
2	2	1	1.324	1.022
1	2	2	1.275	1.012
2	2	2	1.278	1.016
1	3	1	1.288	1.015
2	3	2	1.261	1.016
2	4	2	1.248	1.017

Table 1: Average and worst case ratios for randomly generated instances

The results indicate that the worst-case performance of the Match($3-2\sqrt{2}$)-algorithm improves with increasing number of customers. The highest worst-case ratio is reached when all three sets consist of only one customer. Please note that the competitive ratio is bounded by $\sqrt{2}$ in that case as shown in Lemma 2. Furthermore, the Match($3-2\sqrt{2}$)-algorithm performs well on average. Its solution value is between 1.015 and 1.023 times higher than the optimal offline solution value.

5 Results for the online problem with more than two periods

In this section we consider the problem with more than two periods and investigate whether the results for $T = 2$ can be transferred. Again, the competitive ratio of any algorithm is at most two and the strategies IMMEDIATE - all customers of $C_{t|t+1}$ are served in t for all $t \in \{1, ..., T-1\}$ - and DELAY - all customers of $C_{t|t+1}$ are served in $t+1$ for all $t \in \{1, ..., T-1\}$ - do not perform better than this upper bound in the worst case.

The algorithm SMART(p) can easily be extended to more than two periods. In every period $t \in \{1, 2, ..., T-1\}$ either all customers of set $C_{t|t+1}$ are served (i.e. IMMEDIATELY is applied in t) or all customers in $C_{t|t+1}$ are postponed (i.e. DELAY is applied in t). In the first period the decision of SMART(p) remains the same as in the case of $T = 2$. In period $t \geq 2$ IMMEDIATELY is applied if DELAY was applied in $t-1$ and the distance of serving customers $C_{t-1|t} \cup C_{t|t+1}$ is at most p times the distance needed to serve customers $C_{t-1|t}$. Otherwise, we apply DELAY in period t. In Section 4 we proved that $p = 2$ is the best choice according to the competitive ratio with $CR_{\text{SMART}(2)} = \frac{3}{2}$ for $T = 2$. However, this competitive ratio does not hold for any p for $T > 2$.

Theorem 7. *For $T > 2$ the competitive ratio of SMART(p) is at least 1.6 for any $p \in \mathbb{R}$.*

Proof: Obviously, for $p < 1$ the competitive ratio is two as DELAY is applied in every period for any instance. Hence, we focus on the case of $p \geq 1$ and show the statement for $T = 3$. For this, we consider three different instances in order to receive three lower bounds on the competitive ratio, all depending on p. Afterwards, we determine the value of p that minimizes the maximum of the three lower bounds.

Instance 1: The customers $C_1 = \{v_1\}$, $C_{1|2} = \{v_2, v_3\}$, $C_{2|3} = \{v_4\}$, $C_3 = \emptyset$ are located on a line as shown in Figure 6.

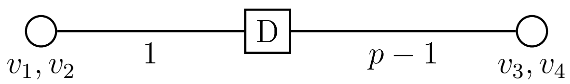

Figure 6: Example of SMART(p) applying IMMEDIATE in $t = 1$

SMART(p) applies IMMEDIATE in period one resulting in a total length of $4p-2$ while the optimal solution has length $2p$. Thus,

$$CR_{\text{SMART(p)}} \geq \frac{4p-2}{2p} = 2 - \frac{1}{p}. \tag{12}$$

Instance 2: The customers $C_1 = \{v_1\}$, $C_{1|2} = \{v_2, v_3\}$, $C_{2|3} = C_3 = \emptyset$ are located on a line as shown in Figure 7.

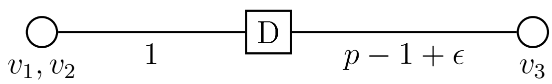

Figure 7: Example of SMART(p) applying DELAY in $t = 1$

SMART(p) applies DELAY in period one resulting in a total length of $2p+2+2\epsilon$ while the optimal solution has length $2p+2\epsilon$. As an arbitrarily small value can be chosen for ϵ, we can conlude

$$CR_{\text{SMART(p)}} \geq \frac{2p+2}{2p} = 1 + \frac{1}{p}. \tag{13}$$

Instance 3: The customers $C_1 = \{v_1\}$, $C_{1|2} = \{v_2, v_3\}$, $C_{2|3} = \{v_4\}$, $C_3 = \{v_5\}$ are located on a line as shown in Figure 8.

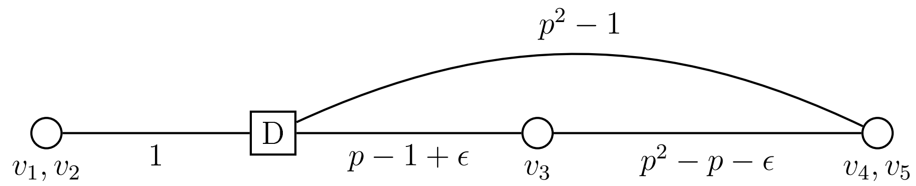

Figure 8: Example of SMART(p) applying DELAY in $t = 1$ and IMMEDIATE in $t = 2$

SMART(p) applies DELAY in $t = 1$ and IMMEDIATE in $t = 2$ resulting in a total length of $4 + 4 \cdot (p^2 - 1)$ while the optimal solution has length $2 + 2 \cdot (p - 1 + \epsilon) + 2 \cdot (p^2 - 1)$. Because of ϵ being arbitrarily small, we conclude

$$CR_{\text{SMART(p)}} \geq \frac{4 + 4 \cdot (p^2 - 1)}{2 + 2 \cdot (p-1) + 2 \cdot (p^2 - 1)} = \frac{2p^2}{p^2 + p - 1}. \tag{14}$$

Hence, we have

$$CR_{\text{SMART(p)}} \geq \max \left\{ 2 - \frac{1}{p}, 1 + \frac{1}{p}, \frac{2p^2}{p^2 + p - 1} \right\}. \tag{15}$$

We look for the value of p that minimizes this term. For $p \in [1, \frac{1+\sqrt{5}}{2}]$ the maximum is attained by function (13) with a minimum of $1 + \frac{2}{1+\sqrt{5}} \approx 1.62$ at the right boundary of the interval. For $p \in [\frac{1+\sqrt{5}}{2}, \frac{3+\sqrt{5}}{2}]$ the maximum is attained by function (14). Deriving this function and determination of its zero point yields a minimum of 1.6 for $p = 2$. For $p \in [\frac{3+\sqrt{5}}{2}, \infty)$ the maximum is attained by function (12) with a minimum of $1 + \frac{2}{1+\sqrt{5}}$ at the left boundary of the interval. By this, the proof is completed. □

Thus, the competitive ratio of the SMART(2) algorithm gets worse for a larger planning horizon. We can even show that - in contrast to $T = 2$ - there is no algorithm with a competitive ratio of 1.5 for $T > 2$ serving all customers of $C_{t|t+1}$ in the same period for all $t \in \{1, ..., T-1\}$.

Theorem 8. *Let A be an online algorithm that decides to serve all customers of $C_{t|t+1}$ in the same period for all $t \in \{1, ..., T-1\}$. Then $CR_A > 1.54$.*

Proof: We consider the case of $T = 3$ with customers $C_1 = \{v_1\}$ and $C_{1|2} = \{v_2, v_3\}$ located on a line as shown in Figure 9 with $x > 0$.

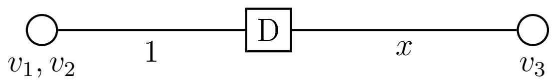

Figure 9: Location of customers in the plane in $t = 1$

There are two possible decisions for algorithm A in $t = 1$.

Case 1: Algorithm A decides to serve customers v_2 and v_3 in $t = 1$.

Let $C_{2|3} = \{v_4\}$ with v_4 located at the same position as v_3 and $C_3 = \emptyset$. Then, algorithm A leads to routes (D, v_1, v_2, D) and (D, v_3, D) in $t = 1$ and (D, v_4, D) in $t = 3$ with a total length of $2 \cdot (1 + 2x)$. However, the optimal routes are (D, v_1, v_2, D) in $t = 1$ and (D, v_2, v_3, D) in $t = 2$ with a total length of $2 \cdot (1 + x)$. Hence, we have

$$CR_A \geq \frac{1 + 2x}{1 + x} =: f_1(x). \tag{16}$$

Case 2: Algorithm A decides to postpone customers v_2 and v_3 to $t = 2$.

Then, we consider the case of $C_{2|3} = \{v_4\}$ with v_4 located as shown in Figure 10 with $y > x$.

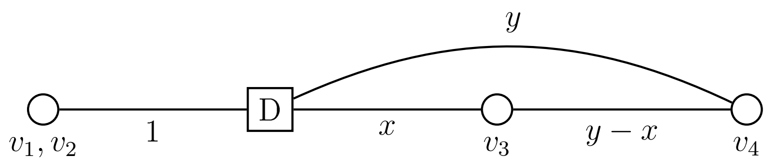

Figure 10: Location of customers in the plane in $t = 2$ in case 2

Again, algorithm A can choose between two strategies:

Case 2a: Algorithm A decides to serve customer v_4 in $t = 2$.

Let $C_3 = \{v_5\}$ with v_5 located at the same position as v_4. Then, algorithm A leads to routes (D, v_1, D) in $t = 1$, (D, v_2, D) and (D, v_3, v_4, D) in $t = 2$ and (D, v_5, D) in $t = 3$ with a total length of $2 \cdot (2 + 2y)$. However, the optimal routes are (D, v_1, v_2, D) in $t = 1$, (D, v_3, D) in $t = 2$ and (D, v_4, v_5, D) in $t = 3$ with a total length of $2 \cdot (1 + x + y)$. Hence, we have

$$CR_A \geq \frac{2 + 2y}{1 + x + y} =: f_2(x, y). \tag{17}$$

Case 2b: Algorithm A decides to postpone customer v_4 to $t = 3$.

Let $C_3 = \emptyset$. Then, algorithm A leads to routes (D, v_1, D) in $t = 1$, (D, v_2, v_3, D) in $t = 2$ and (D, v_4, D) in $t = 3$ with a total length of $2 \cdot (2 + x + y)$. However, the optimal routes are (D, v_1, v_2, D) in $t = 1$ and (D, v_3, v_4, D) in $t = 3$ with a total length of $2 \cdot (1 + y)$. Hence, we have

$$CR_A \geq \frac{2 + x + y}{1 + y} =: f_3(x, y). \tag{18}$$

In summary, we can conclude

$$CR_A \geq \min\{f_1(x), f_2(x, y), f_3(x, y)\}. \tag{19}$$

In order to get the highest possible lower bound, we maximize over x and y. As f_2 is strictly increasing in y and f_3 is strictly decreasing in y with x being fixed, the maximum value of the minimum is reached at equality. Solving the equation

$$\frac{2+2y}{1+x+y} = \frac{2+x+y}{1+y} \tag{20}$$

to y implies

$$y = x - \frac{1}{2} + \sqrt{2x^2 + 2x + \frac{1}{4}}. \tag{21}$$

Inserting into f_2 yields

$$CR_A \geq \min\left\{\frac{1+2x}{1+x}, \frac{1+2x+2\sqrt{2x^2+2x+\frac{1}{4}}}{\frac{1}{2}+2x+\sqrt{2x^2+2x+\frac{1}{4}}}\right\}. \tag{22}$$

The first function is increasing in x, while the second one is decreasing. So, the maximum value of the minimum is attained at equality. Solving this equation to x gives $x \approx 1.1915$ and due to equation (21) $y \approx 3.0308$. Inserting these values into f_1, f_2 or f_3 implies $CR_A > 1.54$. □

When we consider arbitrary algorithms that may split customers of the sets $C_{t|t+1}$, we derive the following lower bound for the competitive ratio.

Theorem 9. *Let A be an arbitrary online algorithm for $T = 3$. Then $CR_A > 1.44$.*

The proof is the same as for theorem 10 in Angelelli et al. (2007). It remains an open question if an algorithm exists that yields this lower bound.

6 Conclusion

We have considered a dynamic vehicle routing problem over multiple periods with strict conditions on the demand of customers and the capacities of the vehicles. The objective of this work was to investigate the competitiveness of different algorithms as well as general lower and upper bounds. As a first result, we observed that the solution of any algorithm has at most twice the length of the optimal solution. Next, we have seen that the SMART(p) algorithm has a competitive ratio of 1.5 for $p = 2$ when the planning horizon is limited to two periods. This is best among all algorithms that serve all customers of set $C_{1|2}$ in the same period. Furthermore, we proved that no algorithm can provide a competitive ratio less than $\sqrt{2}$. Afterwards, we figured out that the competitive ratio of the SMART(p) algorithm increases for an enlarged time horizon. Moreover, there can be no algorithm with competitive ratio $\sqrt{2}$ for more than two periods, as 1.44 is a proven lower bound. These results are summarized in Table 2 with $LB_{I/D}$ denoting the lower bound for algorithms that apply IMMEDIATE or DELAY in every period.

Competitive ratio	LB	$LB_{I/D}$	SMART(p)	UB
$T = 2$	$\sqrt{2}$	1.5	1.5	2
$T \geq 3$	1.44	1.54	≥ 1.6	2

Table 2: Summarized results on competitive ratios

Furthermore, we introduced the Match(λ)-algorithm that reaches the lower bound of $\sqrt{2}$ in the case of $T = 2$ for $\lambda = 3 - 2\sqrt{2}$, at least for a subclass of instances. Yet, it remains to verify this competitive ratio for general instances. Moreover, it is an open question whether this algorithm provides a low competitive ratio for more than two periods. While we detected lower bounds on the competitive ratio for more than two periods, it is not clear if these bounds are tight. From a theoretical point of view it might be interesting to investigate the generalized problem with a capacity of k customers per tour. To approach practical problems the strict conditions on the time windows of the customers must be relaxed. Additionally, it might be interesting to assume probability distributions for future customer requests and minimize the expected costs by anticipatory strategies.

Bibliography

Albareda-Sambola, M., Fernández, E., and Laporte, G. (2014). The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers & Operations Research*, 48:31–39.

Angelelli, E., Bianchessi, N., Mansini, R., and Speranza, M. G. (2009). Short term strategies for a dynamic multi-period routing problem. *Transportation Research Part C: Emerging Technologies*, 17(2):106–119.

Angelelli, E., Speranza, M. G., and Savelsbergh, M. W. P. (2007). Competitive analysis for dynamic multiperiod uncapacitated routing problems. *Networks*, 49(4):308–317.

Archetti, C., Mansini, R., and Speranza, M. (2005). Complexity and reducibility of the skip delivery problem. *Transportation Science*, 39(2):182–187.

Asano, T., Katoh, N., Tamaki, H., and Tokuyama, T. (1997). Covering points in the plane by k-tours: Towards a polynomial time approximation scheme for general k. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, pages 275–283. ACM.

Bazgan, C., Hassin, R., and Monnot, J. (2005). Approximation algorithms for some vehicle routing problems. *Discrete Applied Mathematics*, 146(1):27–42.

Billing, C., Jaehn, F., and Wensing, T. (2018). A multiperiod auto-carrier transportation problem with probabilistic future demands. *To appear in Journal of Business Economics.*

Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group.

Edmonds, J. (1965). Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards*, 69(1-2):125–130.

Haimovich, M. and Rinnooy Kan, A. H. G. (1985). Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542.

Haimovich, M., Rinnooy Kan, A. H. G., and Stougie, L. (1988). Analysis of heuristics for vehicle routing problems. In Golden, B. and Assad, A., editors, *Vehicle Routing: Methods and studies*, volume 16 of *Studies in Management Science and Systems*, pages 47–61. North-Holland.

Hassin, R. and Rubinstein, S. (2005). On the complexity of the k-customer vehicle routing problem. *Operations Research Letters*, 33(1):71–76.

Karlin, A. R., Manasse, M. S., Rudolph, L., and Sleator, D. D. (1988). Competitive snoopy caching. *Algorithmica*, 3(1-4):79–119.

Larsen, A., Madsen, O. B. G., and Solomon, M. M. (2008). Recent developments in dynamic vehicle routing systems. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43, pages 199–218. Springer, Boston.

Li, C.-L. and Simchi-Levi, D. (1990). Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems. *ORSA Journal on Computing*, 2(1):64–73.

Ritzinger, U., Puchinger, J., and Hartl, R. F. (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231.

Sleator, D. D. and Tarjan, R. E. (1985). Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208.

Toth, P. and Vigo, D., editors (2002). *The Vehicle Routing Problem.* Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia.

Wen, M., Cordeau, J.-F., Laporte, G., and Larsen, J. (2010). The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, 37(9):1615–1623.

IV Beitrag B3

Fair Task Allocation Problem

Pre-Print of: Billing, C., Jaehn, F., and Wensing, T. (2020). Fair Task Allocation Problem. *Annals of Operations Research*, 284:131-146.
https://doi.org/10.1007/s10479-018-3052-3

Abstract

In fields like transport or materials sourcing, it is common industrial practice nowadays to contract several partners for the fulfilment of similar sets of tasks. A typical approach is to include quotes to the contracts that specify which portion of the total volume should be given to each partner. In this study, which is inspired by a real-world problem, we examine the question of operationally distributing jobs to a set of partners in order to meet the contracted quotes in different dimensions as closely as possible. We propose the term Fair Task Allocation Problem (FTAP) and analyze its complexity. While the problem is NP-hard in the strong sense for the general case, we show that it is solvable in pseudopolynomial time for a given number of partners and dimensions. Besides an exact solution approach based on dynamic programming, we present an efficient Tabu Search procedure. The Tabu Search is applied to real world as well as to self-generated instances. To verify its quality, the results are compared to the solutions of a commercial MIP-solver.

1 Introduction

The real-world problem addressed in this study is the following. A manufacturing company employs contract hauliers with the shipment of their products. The transport volumes are combined to truck loads in advance and the resulting trips are then assigned to the hauliers for execution. Since in many countries the logistics sector is dominated by dedicated logistics specialists that offer competitive freight rates, this is common practice for manufacturing companies. The strategy to employ different partners with the same kind of jobs (multi-sourcing) is mainly followed to avoid negative consequences of having to rely on a single partner. Vice versa, a haulier may also be reluctant to sign a high-volume contract and dedicate significant fractions of its capacity to one single customer.

There are two reasons why companies seek automated decision support in assigning jobs to contracted partners. Firstly, the assignment process may be improved in terms of fairness and contract fulfilment, which potentially leads to less complaints from the partners. Secondly, manual assignments come with the risk of corruption, since there are typically at least some beneficial or unfavourable aspects of trips that are not covered by the contracted payments so that hauliers may be tempted to approach the dispatchers in order to receive more convenient trips. In particular, the risk of corruption should be avoided as far as possible in the interest of the focal company of a supply chain. This is even more important if it concerns a global supply chain that includes partners in less developed countries with a high susceptibility to corruption.

In general terms, the considered problem can be described as follows. We have to distribute n jobs to m agents that should execute them. Each job has numerical properties in different dimensions and every agent has target values for each dimension that should be met as closely as possible. To be more precise, we want to minimize the sum of deviations of the target values from the sum of the numerical properties of the assigned jobs multiplied with a dimension-dependent weight over all agents and dimensions.

In our real-world problem, the target values are computed according to percentage target quotes for each dimension involved. On each assignment occasion the target quotes should be met as closely as possible on a cumulative basis. Thus, it is possible that target quotes differ from day to day and also throughout the dimensions in order to balance out cumulative deviations that cannot be completely avoided.

As an example, let us consider three trips (jobs) of 100, 75, and 25 kilometers that should be distributed to two hauliers (agents). Given equal target quotes of 50% in the dimensions *number of trips* and *kilometers*, both hauliers should receive 1.5 trips with a total length of 100 kilometers. Obviously, assuming equal weights in both dimensions, we cannot do better than assigning the 100-km-trip to the one and the other two trips to the other haulier, which perfectly meets the requirement in the dimension *kilometer*, but creates an imbalance in the dimension *number of trips*. On the next planning occasion, we should thus seek to again distribute kilometers evenly but try to assign one trip more to the first haulier to resolve the imbalance in that dimension.

A company that has to solve such assignment problems on a regular basis may probably find it most convenient to embed a proper solution routine into the software system used in the according field of application. The benefits of meeting contracted quotes more closely and treating the agents more fairly in the long term are not easily monetized. Agents that continuously feel being treated in an unfair way will be dissatisfied and may seek to dissolve the contract, while they will hardly notice that the distribution of jobs has been less fair than it could have been on a single planning occasion. Besides the technical requirement of well meeting the target quotes, there is therefore an economic requirement to the solution method that it can be implemented and maintained at moderate costs.

Apart from the haulier context described above, this problem may arise in various fields when a pool of different tasks must be assigned to multiple workers. Here, the assignment of shifts or tasks to workers can lead to dissatisfaction if done in an unfair way. Concretely, the problem arises at airports (or other large facilities) in the rostering of ground workers. Typically, these workers are qualified for a variety of different tasks ranging from indoor activities, e.g., at baggage belts, to outdoor activities, such as loading and unloading airplanes or maintaining roads and runways especially in rainy or cold periods of the year. The various kinds of tasks have to be covered by shifts that are then taken by workers during the airport's operative hours. Apart from a shift's basic properties, such as the day in the week, its total hours, and payment, it can have other fairness-relevant characteristics, such as (general) task attractiveness, associated physical exertion or the exposure to noise or extreme weather conditions. The target quotes of the workers may vary because of different contracts or personal wishes.

The problem is about establishing fairness and eventually reducing corruption in a job assignment process so that it falls under social sustainability (see Jaehn (2016)). It is related to generalized assignment problems (GAP), which are intensely studied in the literature since the famous seminal work of Kuhn (1955). See Cattrysse and Van Wassenhove (1992) or Pentico (2007) for surveys. While such problems typically aim at the minimization of the overall assignment costs, Mazzola and Neebe (1988) formulate two bottleneck versions of the generalized assignment problem (BGAP). The Task BGAP (Mazzola and Neebe (1993), Martello and Toth (1995a), and Martello and Toth (1995b)) demands the maximum costs of assigning a task to an agent to be minimized, while the Agent BGAP minimizes the maximum total assignment costs of an agent. Karsu and Azizoğlu (2012) investigate the Agent BGAP if the costs correspond to the consumption of resources. The bottleneck version of GAP relates closest to our problem. In contrast, we consider the different objective function of minimizing the sum of deviations from individual target levels, where there are no individual assignment costs c_{ij} incurred if task i is allocated to agent j.

If only one dimension is of interest, our problem is similar to multi-way number partitioning, where a given set of integers must be divided into subsets such that the sums of these subsets are as close to each other as possible. Two exact algorithms to minimize the largest subset sum are presented by Korf (2009). In a subsequent paper, Korf (2010) states that these algorithms can also be used to maximize the smallest subset sum, but they fail to minimize the difference between the largest and the smallest subset. This difference is minimized by heuristic algorithms of Zhang et al. (2011) for the balanced multi-way number partitioning problem, where additionally the difference of cardinality must be as small as possible among the subsets. In minimizing the sum of deviations and allowing diverse target values, our problem differs from multi-way number partitioning.

In Kubiak (2009), several problems regarding fairness are considered, e.g., the apportionment problem arising from the distribution of seats in a parliament Another well studied problem is the car sequencing problem, first mentioned by Parrello et al. (1986), that looks for a feasible sequence to meet the capacity constraints of an assembly line. Very similar to this is the Product Rate Variation problem, see for example Józefowska (2012), that arises in just-in-time production planning. Kubiak and Sethi (1991) show that it is solvable in polynomial time by reduction to an assignment problem for a subclass of objective functions. The balancing of schedules in a multi-level just-in-time assembly system in order to reduce inventories is considered in Kubiak et al. (1997), who present a nonlinear integer program with a min-max objective function to balance the production of various products according to given demand ratios. The same problem with a min-sum objective function was investigated by Miltenburg and Sinnamon (1989).

We will work out that our problem in a one dimensional setting is also related to scheduling problems on parallel machines. Identical target values can be seen as common due dates and an assignment that exceeds the target values corresponds to late work in the scheduling environment. See Sterna (2011) for a survey on scheduling problems with late work criterion. There is no literature for minimizing late work on parallel machines with common due dates, but we will see that there is also a strong relation to the more investigated problem of minimizing tardiness. Using the notation established by Graham et al. (1979), $Pm|d_j = d|\sum T_j$, which is shown to be NP-hard by Kovalyov and Werner (2002), is similar to the problem considered here. Kovalyov and Werner (2002) also show that the problem is NP-hard in the strong sense if m is part of the input and they give fully polynomial time approximation schemes for $Pm|d_j = d|\sum T_j + d$. Huynh Tuong et al. (2009) consider the more general problem $Pm|d_j = d|\sum w_jT_j$, for which they present an exact pseudopolynomial algorithm.

Although our problem shows similarities to several problems of different fields of research, none of them fits completely to our real-world problem (to the best of our knowledge). The main reason for this is the fact that we are interested in a fair distribution in several dimensions. So, our contribution to the existing literature is the investigation of a new problem. As we choose a quite general formulation, the use of our study is not limited to the real-world problem described above. Rather, it can be useful for many applications in which a fair distribution is searched for and fairness is measured according to several properties.

The article is organized as follows. We will start with a formal definition of the considered problem in section 2. Afterwards, we present some properties of the problem in section 3, which lead to scheduling formulations for a special case of the problem. In section 4, we analyze the complexity of the problem. Following this, in section 5, we present a Tabu Search algorithm and an exact algorithm that runs in pseudopolynomial time if the number of agents and dimensions is fixed. Results of some numerical tests are presented in section 6, while section 7 concludes the study.

2 Problem definition

Let T be a set of tasks with $|T| = n$ and A be a set of agents with $|A| = m$ that should execute them. Furthermore, we consider a set of dimensions denoted by D. Each task t has a numerical property $p_{t,d} \in \mathbb{N}$ in every dimension d and each agent has target values $u_{a,d} \in \mathbb{R}$ throughout the dimensions that should be met as closely as possible by the assignments. Additionally, we introduce positive weights w_d to make the deviations of the dimensions comparable to each other. The objective is to minimize the sum of the weighted absolute deviations of the allocation for every agent in every dimension to the corresponding target values, where each task has to be assigned to exactly one agent.

Fair Task Allocation Problem (FTAP)

The binary decision variable $y_{t,a}$ denotes whether task t is assigned to agent a ($y_{t,a} = 1$) or not ($y_{t,a} = 0$). Let $x_{a,d}$ be the utilization of agent a in dimension d and $\delta_{a,d}$ be the corresponding absolute deviation from the agent's target in this dimension. Thereby, the problem can be modeled as follows:

$$\text{Min} \sum_{d \in D} \sum_{a \in A} w_d \cdot \delta_{a,d} \tag{1}$$

subject to

$$\sum_{a \in A} y_{t,a} = 1 \qquad \forall t \in T \tag{2}$$

$$\sum_{t \in T} y_{t,a} \cdot p_{t,d} = x_{a,d} \qquad \forall a \in A, d \in D \tag{3}$$

$$\delta_{a,d} \geq x_{a,d} - u_{a,d} \qquad \forall a \in A, d \in D \tag{4}$$

$$\delta_{a,d} \geq u_{a,d} - x_{a,d} \qquad \forall a \in A, d \in D \tag{5}$$

$$y_{t,a} \in \{0, 1\} \qquad \forall a \in A, t \in T \tag{6}$$

$$x_{a,d} \in \mathbb{N} \qquad \forall a \in A, d \in D \tag{7}$$

$$\delta_{a,d} \in \mathbb{R} \qquad \forall a \in A, d \in D \tag{8}$$

The objective (1) is to minimize the sum of all weighted deviations. Constraints (2) ensure that every task is assigned to exactly one agent. Constraints (3) guarantee $x_{a,d}$ to be the utilization of agent a in dimension d. In an optimal solution, constraints (4) [(5)] are tight if agent a receives too much [too little] in dimension d. The domains of the variables are given by constraints (6), (7), and (8).

FTAP with type restrictions

A practically relevant extension of the FTAP is the introduction of types of tasks, where each agent can only accept a limited number of tasks for each type. In our real-world problem, for example, the trips may require certain truck types of which each haulier may only provide a limited amount. Let K be the set of different types and $c_{a,k}$ the number of tasks of type k that agent a can execute. Moreover, the parameter $z_{t,k}$ indicates whether task t is of type k ($z_{t,k} = 1$) or not ($z_{t,k} = 0$). Every task belongs to exactly one type, i.e., $\sum_{k \in K} z_{t,k} = 1 \ \forall t \in T$ holds. Then we can introduce the capacity restrictions by adding the constraints

$$\sum_{t \in T} z_{t,k} \cdot y_{t,a} \leq c_{a,k} \quad \forall a \in A, k \in K \tag{9}$$

to the above model. We refer to the model with constraints (9) as the Restricted Fair Task Allocation Problem (RFTAP).

Alternative objective function

Objective (1) seeks to minimize the total imbalance, which may induce that most agents receive a convenient load, while some few have to arrange with high deviations. In applications with consecutive decisions as in our real-world problem, these high deviations can be balanced at subsequent planning occasions. However, if we face a unique decision, it might be more reasonable to minimize the maximum deviation that any agent will observe.

We can model this by replacing objective (1) by (10) and adding constraints (11) and (12) to the model.

$$\text{Min } \Delta \tag{10}$$

$$\Delta \geq \sum_{d \in D} w_d \cdot \delta_{a,d} \quad \forall a \in A \tag{11}$$

$$\Delta \in \mathbb{R} \tag{12}$$

The following example illustrates that the two alternative objectives may indeed lead to different optimal solutions even in the one-dimensional case. Let us consider seven tasks with volumes $\{5, 10, 12, 15, 17, 18, 23\}$ that must be assigned to four agents with target values $\{20, 20, 30, 30\}$. Figure 1 shows the optimal solutions for both objective functions.

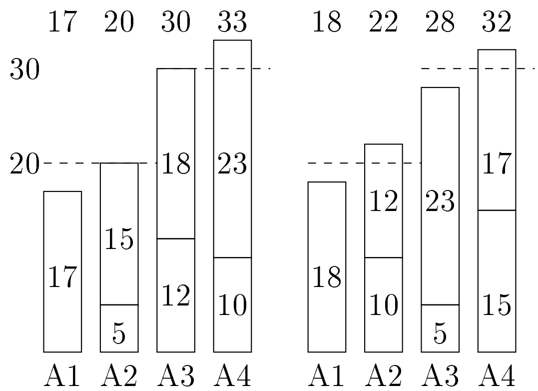

Figure 1: Best solution for objective function (1) and (10)

In order to stay close to our specific application, we will focus on the study of objective function (1) here.

3 Properties and related scheduling problems

In this section, we present some selected properties of FTAP. As a consequence, we will derive alternative formulations for special cases of FTAP with one dimension, which are known from scheduling theory. By this, we point out a practical application of theoretical scheduling problems. Additionally, we will be able to strengthen complexity results of these scheduling problems in the next section. For $x \in \mathbb{R}$, let us denote $x^+ = \max\{x, 0\}$.

Property 1. *The objective function of FTAP can be replaced by*

$$Min \sum_{d \in D} \sum_{a \in A} w_d \cdot (x_{a,d} - u_{a,d})^+,$$

i.e., the minimization of the weighted deviation above the targets is equivalent to the minimization of the weighted overall deviation.

This property holds due to the fact that the minimization of the amount that is assigned above the target value also leads to the minimization of the lacking amount. Assignments exceeding the targets can be regarded as late work referring to scheduling theory. Thus, using the notation established by Graham et al. (1979), we can conclude that FTAP in one dimension with m agents and equal target values is equivalent to $Pm|d_j = d, \sum_j p_j = m \cdot d|Y$. In that problem, a set of jobs $J = \{1, \ldots, n\}$ with integer processing times p_j $\forall j \in J$ must be scheduled on m identical parallel machines. Each job has a common due date d and we additionally require $\sum_j p_j = m \cdot d$. At any time, a machine can handle only one job. The objective is to minimize the late work, i.e., minimize $Y = \sum_j Y_j = \sum_j \min\{p_j, T_j\}$ with $T_j = \max\{C_j - d_j, 0\}$, where C_j denotes the completion time of Job j.

The FTAP in one dimension with equal targets is equivalent to this problem. The numerical properties p_t correspond to processing times p_j in the scheduling environment and the agents can be regarded as machines. All jobs have a common due date d, which is equal to the common target value u. Therefore, in contrast to most scheduling problems, the sequencing of the jobs on a machine has no influence on the objective function, which can be written as $Y = \sum_k \max\{C^k_{\max} - d, 0\}$, when $C^k_{\max}$ denotes the completion time of the last job on machine k. Additionally considering Property 1, the equivalence becomes obvious.

We argue that in many cases, the sum of the targets corresponds to the sum of the properties. Especially if the targets are originally given in percent and then converted into absolute values, the assumption $\sum_{t\in T} p_{t,d} = \sum_{a\in A} u_{a,d}\ \forall d \in D$ is justified. Hence, from now on we assume this equality. Thereby, we obtain two more useful properties.

Property 2. *For every - not necessarily optimal - solution of FTAP, the following equation holds:*

$$\sum_{a\in A}(x_{a,d} - u_{a,d})^+ = \sum_{a\in A}(u_{a,d} - x_{a,d})^+ \ \forall d \in D.$$

Property 3. *Every optimal solution of FTAP with one dimension has the following property:*

$$x_a - u_a < \inf\{p_t | t \in T, y_{t,a} = 1\} \ \forall a \in A.$$

We use the infimum instead of the minimum as the set $\{p_t | t \in T, y_{t,a} = 1\}$ could be empty. The property can be verified by a contradiction argument. If there was a task t assigned to agent a with $p_t \leq x_a - u_a$, due to Property 2 there must be an agent a' with $x_{a'} - u_{a'} < 0$. Hence, assigning this task t to agent a' reduces the overall deviation.

If we transfer Property 3 to the scheduling environment, we can conclude that in an optimal solution of $Pm|d_j = d, \sum_j p_j = m \cdot d|Y$, not more than one job can be tardy on each machine. This leads to $Y_j = T_j\ \forall j \in J$ and, therefore, FTAP in one dimension with m agents and equal target values is also equivalent to $Pm|d_j = d, \sum_j p_j = m \cdot d| \sum_j T_j$.

Corollary 1. *FTAP in one dimension with* m *agents and equal target values is equivalent to* $Pm|d_j = d, \sum_j p_j = m \cdot d| \sum_j T_j$.

4 Complexity results for the FTAP

For analyzing the complexity of the FTAP we will use the following NP-complete decision problems. They are almost literally taken from Garey and Johnson (1979).

3-Partition
Instance: A finite set $A = \{a_1, a_2, \ldots, a_{3m}\}$ of $3m$ elements with size $s(a_i) \in \mathbb{N}\ \forall i \in \{1, 2, \ldots, 3m\}$ and a bound $B \in \mathbb{N}$ such that $B/4 < s(a_i) < B/2\ \forall i \in \{1, 2, \ldots, 3m\}$ and $\sum_{i=1}^{3m} s(a_i) = mB$.

Question: Can A be partitioned into m disjoint sets $S_1, \ldots, S_m$ such that $\sum_{a \in S_j} s(a) = B\ \forall j \in \{1, 2, \ldots, m\}$?

Exact cover by 3 sets (X3C)
Instance: A finite set X with $|X| = 3 \cdot q$ for a $q \in \mathbb{N}$ and a collection $\mathcal{C}$ of 3-element subsets of X.

Question: Does $\mathcal{C}$ contain a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ such that every element of X occurs in exactly one member of $\mathcal{C}'$?

Partition
Instance: A finite set $A = \{a_1, a_2, \ldots, a_n\}$ with size $s(a_i) \in \mathbb{N}\ \forall i \in \{1, 2, \ldots, n\}$.

Question: Is there a set $I \subseteq \{1, 2, \ldots, n\}$ with $\sum_{i \in I} s(a_i) = \sum_{i \notin I} s(a_i)$?

3-Partition was shown to be NP-complete in the strong sense, see (Garey and Johnson, 1979, pp.96ff). The same complexity result holds for X3C (Garey and Johnson, 1979, p.53). The Partition problem was identified to be NP-complete in the ordinary sense by Karp (1972). By reduction of these problems, we will prove the following theorem:

Theorem 1. *1. FTAP is NP-hard in the strong sense even for a constant number of dimensions.*

2. FTAP is NP-hard in the strong sense even for a constant number of agents.

3. FTAP is NP-hard in the ordinary sense for a constant number of dimensions and agents.

Proof: We will verify the three statements by polynomial reductions of the above problems to the decision problem "Is there a solution of FTAP with deviation 0?".

1. We consider an arbitrary instance of 3-Partition and construct an instance of FTAP with one dimension and m agents. The target values u_a are set equal to B for all agents and the properties of the $3m$ tasks are defined as $p_t = s(a_t)\ \forall t \in \{1, 2, \ldots, 3m\}$. Note that the weights can be chosen arbitrarily as we are looking for an assignment without any deviation. If there exists an allocation without any deviation, every agent obtains three tasks as $u_a/4 < p_t < u_a/2\ \forall t \in \{1, 2, \ldots, 3m\}$ holds. So, if we put the tasks of agent i in the set S_i, we receive a "Yes-certificate" for 3-Partition. On the other hand, if any allocation of the tasks leads to a deviation, the answer of 3-Partition is "No".

2. We take an instance of X3C and assume $X = \{1, 2, \ldots, 3q\}$ without loss of generality. As X is a finite set, we can conclude that $\mathcal{C} = \{\{x_{1,1}, x_{1,2}, x_{1,3}\}, \{x_{2,1}, x_{2,2}, x_{2,3}\}, \ldots, \{x_{n,1}, x_{n,2}, x_{n,3}\}\}$ must also be finite. For notational convenience, we define $C_t = \{x_{t,1}, x_{t,2}, x_{t,3}\}\ \forall t = 1, 2, \ldots, n$ and assume that every $x \in X$ is included in at least one set C_t as otherwise the answer of X3C is obviously "No". We construct an instance of FTAP with two agents, $3q$ dimensions, and arbitrary weights. The targets for the first agent are set equal to one in all dimensions: $u_{1,d} = 1\ \forall d \in \{1, 2, \ldots, 3q\}$. The targets of the second agent are given by the following equation:

$$u_{2,d} = |\{C_t | t \in \{1, 2, \ldots, n\} \wedge d \in C_t\}| - 1 \quad \forall d \in \{1, 2, \ldots, 3q\}$$

We define n tasks, so every subset $C_t \in \mathcal{C}$ corresponds to a task. The numerical property of a task t is set to one in a dimension d if $d \in C_t$ holds and is set to zero in all other dimensions, i.e.:

$$p_{t,d} = 1, \text{ if } d \in C_t$$
$$p_{t,d} = 0, \text{ otherwise.}$$

Assume that an allocation of tasks exists leading to zero deviation, so we conclude that the first agent receives exactly one unit in every dimension. Let T_1 denote the set of tasks assigned to the first agent. Then, the answer for X3C is "Yes", since we can put all sets C_t with $t \in T_1$ in the collection $\mathcal{C}'$. On the other hand, if we have a "Yes"-instance of X3C, let $T_1 \subseteq \{1, 2, \ldots, n\}$ denote the indices of the subsets included in $\mathcal{C}$, i.e., $C_t \in \mathcal{C} \iff t \in T_1$. Then, we assign the tasks included in T_1 to the first agent and the other tasks to the second agent. By this assignment, the first agent receives one unit per dimension and, hence, zero deviation is obtained. Note that all numerical values are polynomially bounded in the input length of X3C, especially $u_{2,d} \leq n \leq (3q)^3 \; \forall d \in \{1, 2, \ldots, 3q\}$.

3. This is shown by reduction of Partition, similar to the first part of this proof. Therefore, consider two agents in a one dimesional setting with target values $u_a = \sum_{i=1}^{n} s(a_i)/2$ for both agents and the properties of the n tasks are set to $p_t = s(a_t) \; \forall t \in \{1, 2, \ldots, n\}$. Again, the weights can be chosen arbitrarily. □

The proofs of part one and three of Theorem 1 construct instances of FTAP with one dimension and equal target values. Thus, keeping in mind the equivalence of FTAP in one dimension and equal target values to $Pm|d_j = d, \sum_j p_j = m \cdot d|Y$ we obtain the following complexity results.

Corollary 2. *1. $P|d_j = d, \sum_j p_j = m \cdot d|Y$ is NP-hard in the strong sense.*

2. $P2|d_j = d, \sum_j p_j = m \cdot d|Y$ is NP-hard.

This strengthens the findings of Chen et al. (2016), who proved the same complexity results for the corresponding problems without the condition $\sum_j p_j = m \cdot d$.

For the RFTAP, we obtain the same complexity results. This can be verified by setting $c_{a,k} = n \; \forall a \in A, k \in K$. Thereby, every assignment is feasible and the above proofs remain true. Moreover, these complexity results also hold for the alternative objective function (10) presented in section 2 as the proofs stay the same.

5 Algorithms for the FTAP

An obvious solution strategy to the FTAP is to model the problem as a Mixed Integer Program (MIP) and apply a commercial MIP solver. For real-world problems, this may be inappropriate for two reasons. Firstly, since the FTAP is an NP-hard problem, solvers may fail to provide a solution in reasonable runtime. Secondly, the benefits of solving the problem are not easily monetized so that it may be difficult to economically justify the purchase and maintenance of a MIP solver licence for the given purpose. Particularly, this is the case if there already exists a software system in the according field of application that is not equipped with a proper licence. It may then be more reasonable also from a budget perspective to invest a couple of developer days to implement a problem-specific method. In this section, we therefore develop a Tabu Search to solve the problem and show that it is competitive with state-of-the-art MIP solution techniques. Moreover, we outline a dynamic programming approach that is able to compute exact solutions in pseudopolynomial time.

Tabu Search

Tabu Search is a metaheuristic that can be applied to many combinatorial optimization problems, see Glover (1989). In its original form, it works as follows. In a first step, an initial solution is determined. Starting from this, a neighborhood with similar solutions is built up by some predefined moves. To avoid being captured in a local optimum, some moves are set tabu for a specific number of iterations. The best neighbor obtained by a non-tabu-move is chosen as the current solution and the tabu list is updated. Additionally, if the current solution is better than the previously best known one, it is stored as the best solution so far. Furthermore, a termination criterion must be defined, for example, a fixed number of iterations or a specific value of the objective function that must be reached.

In our implementation of the Tabu Search, we generate an initial solution by a random procedure. Therefore, we compute the average relative target of agent a by

$$pr(a) = \frac{1}{|D|} \cdot \sum_{d \in D} \frac{u_{a,d}}{\sum_{t \in T} p_{t,d}}.$$

The value $pr(a)$ is used as the probability for assigning a task to agent a. So, to get an initial solution, the tasks are randomly distributed to the agents according to these probabilities. The neighborhood of a solution is built up by the three different operations *change_agent*, *exchange_tasks*, and *loop_exchange*.

1. *change_agent*: Assign task t_1 to a different agent. This leads to $n \cdot (m-1)$ new solutions, since we do this for all tasks and all possible new agents.

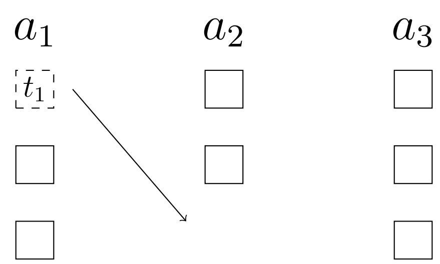

Figure 2: Procedure *change_agent*

2. *exchange_tasks*: Select tasks t_1 and t_2 from agents a_1 and a_2, $a_1 \neq a_2$ and exchange their agents. By doing this for every pair of tasks, we get new solutions in the order of n^2.

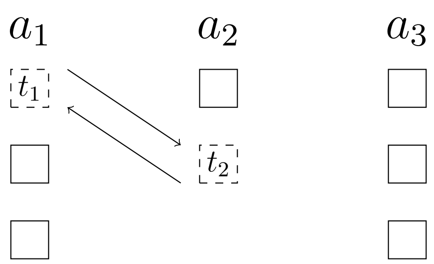

Figure 3: Procedure *exchange_ tasks*

3. *loop_ exchange*: Here we first choose a task t_1 and let a_1 be the agent that executed this task. We break this assignment and look for another task t_2 to assign to agent a_1. The task t_2 is determined in such a way that the sum of deviations over all dimensions of agent a_1 is minimized. Assume that t_2 has been assigned to agent a_2 before. Then, in the last step, we determine the task t_3 assigned to another agent a_3 such that the deviation of agent a_2 and a_3 is minimized when they are assigned to task t_3 and t_1, respectively. As we execute the *loop_ exchange* for every possible task t_1, we create n new solutions, where we need to check up to $n + n$ candidates for t_2 and t_3.

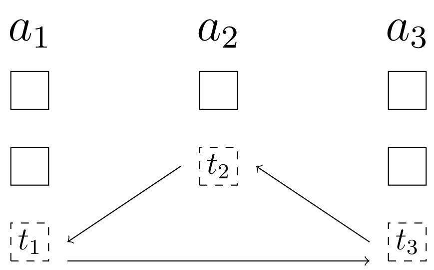

Figure 4: Procedure *loop_ exchange*

Our tabu list consists of task-agent pairs. If a task t_1 that was assigned to agent a_1 changes the agent in a step of the Tabu Search, the pair (t_1, a_1) is set tabu. Thus, in the next *tabu_ length* iterations, task t_1 cannot be assigned to agent a_1 again. The value of *tabu_ length* is fixed at the beginning of the algorithm. If the new solution arises from the *change_ agent* operation, we receive one new element in the tabu list. If the best neighbor originates from the *exchange_ tasks* procedure, two pairs are added to the tabu list, and if the *loop_ exchange* leads to the new solution, three new pairs are set tabu.

As mentioned before, we look for the best neighbor that is not tabu and choose it as the *solution*. As an exception, we disregard the tabu status of a neighborhood solution if it is better than the *best solution* so far. This is the most commonly used aspiration criterion, see Gendreau and Potvin (2014). After the selection of the *current solution* the tabu list is updated. If the best solution has not been improved over the last l iterations, we execute an *intensification_ phase*. In that phase, we continue the Tabu Search with the best solution as starting point and an empty tabu list. Furthermore, we determine the $\lceil m/2 \rceil$ agents with the lowest deviations from targets in the best solution and freeze the assignment of their tasks. As the neighborhood is reduced by this, we use a smaller tabu length in the *intensification_ phase*, which is stopped after a predefined number of iterations. The *overall best solution* is updated if necessary.

After the *intensification_ phase*, we restart the Tabu Search with an empty tabu list to diversify the search space. For the initial solution of the restart, we apply the procedure *new_ start_ solution*. The agent with the lowest deviation from its targets in the *best solution* of the last run keeps its tasks, whereas all other tasks are randomly assigned to the other agents according to their average relative targets $pr(a)$.

The algorithm terminates after r restarts or as soon as a limit of i iterations is reached not including the iterations of the *intensification_ phase*. The overall procedure is summarized in the pseudocode of Algorithm 1. For notational convenience, we write $f(X)$ for the objective function value of solution X. Moreover, X_{best} denotes the *best solution* of a restart and X^* the *overall best solution*. Furthermore, TL denotes the tabu list.

Algorithm 1: Tabu Search

Input: l (number of iterations without improvement), r (maximum number of restarts), i (maximum number of iterations)

Construct an initial solution X_{init} by randomly assigning all tasks to the agents according to the probabilities $pr(a) = \frac{1}{|D|} \cdot \sum_{d \in D} \frac{u_{a,d}}{\sum_{t \in T} p_{t,d}}$ $\forall a \in A$;

Set $X_{cur} = X_{init}$, $X_{best} = X_{init}$, $X^* = X_{init}$, $TL = \emptyset$, $count = 0$;

```
while i > 0 and r > 0 do
    count++;
    Determine neighborhood N(X_cur) by execution of change_agent,
      exchange_tasks, and loop_exchange;
    Determine tabu neighbors TN(X_cur);
    X_cur = argmin{f(X)|X ∈ N(X_cur) \ TN(X_cur)};
    if min{f(X)|X ∈ TN(X_cur)} < min{f(X_cur), f(X_best)} then
        X_cur = argmin{f(X)|X ∈ TN(X_cur)};
    end if
    update TL;
    if f(X_cur) < f(X_best) then
        X_best = X_cur;
        count = 0;
        if f(X_best) < f(X*) then
            X* = X_best;
        end if
    end if
    if count > l then
        X_best = intensification_phase(X_best);
        r = r − 1;
        if f(X_best) < f(X*) then
            X* = X_best;
        end if
        X_cur = new_start_solution(X_best);
        TL = ∅;
        count = 0;
    end if
    i = i − 1;
end while
return X*
```

Exact algorithm

Our algorithm is based on the concept of dynamic programming. We will not focus on the assignment of the tasks to the agents but on the resulting allocation in every dimension, similar to the approach in Kubiak et al. (1997). Therefore, we store a (partial) solution in a $|A| \times |D|$-matrix $X = (x_{a,d})_{a \in A, d \in D}$, in which every row represents an agent and every column represents a dimension. The value $x_{a,d}$ holds the current allocation of agent a in dimension d. The set of all possible matrices X after the assignment of the first t tasks is given by S^t, so S^n contains all final solutions. Let $P = \max_{t \in T, d \in D} p_{t,d}$. Then, it is easy to see that $|S^t| \leq (t \cdot P)^{|A| \cdot |D|}$ holds for all $t \in \{1, \ldots, n\}$ and, therefore, the number of different solutions is bounded by $(n \cdot P)^{|A| \cdot |D|}$. This delivers a pseudopolynomial algorithm if the number of agents and the number of dimensions is fixed. However, for realistic values of $|A|$ and $|D|$, the running time of this procedure is very high, so we excluded it from our computational study.

In a first step, we compute a good solution by a heuristic, e.g., by the Tabu Search proposed before. Hence, we get an upper bound UB, which can be used to eliminate partial solutions, so we obtain a bounded dynamic programming algorithm. We presume an arbitrary order of the tasks in set T. In step t based on the set S^{t-1}, we generate $|A| \cdot |S^{t-1}|$ new partial solutions by assigning task t to every agent starting from every matrix in the set S^{t-1}. By eliminating identical partial solutions, we receive the set S^t. Furthermore, partial solutions with $\sum_{a,d} w_d \cdot \max\{0, x_{a,d} - u_{a,d}\} > UB/2$ cannot lead to an optimal assignment, so these matrices are also deleted. This procedure continues until we obtain S^n.

Figure 5 demonstrates the possibility that two different assignments can result in the same partial solution, so one of them is discarded. Therefore, consider the following example with two agents, two dimensions, and numerical properties of the first three tasks $p_1 = (1, 3), p_2 = (2, 8), p_3 = (1, 5)$.

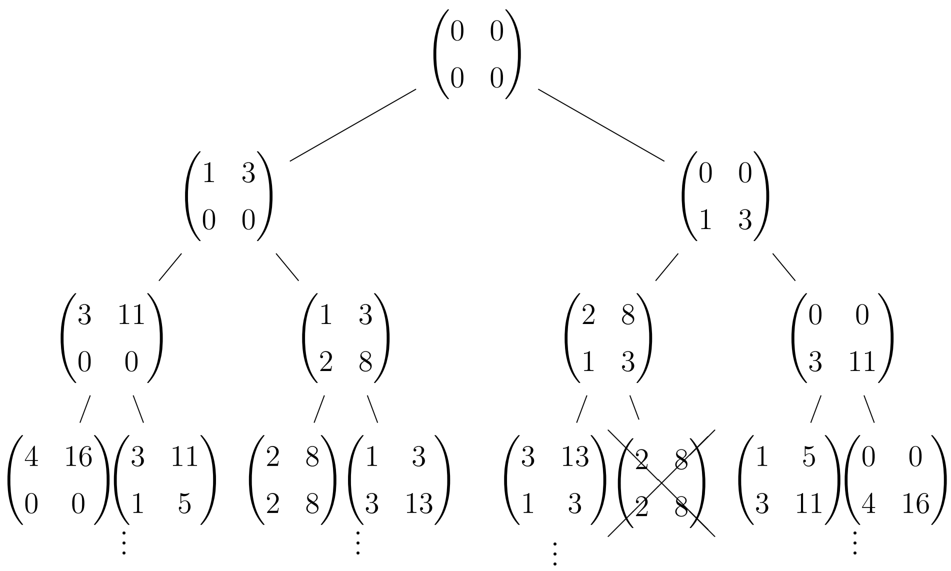

Figure 5: Different assignments that lead to equal utilizations

The whole procedure is shown in the pseudocode of Algorithm 2. Again, the objective function value of a solution X is given by $f(X)$.

This algorithm can easily be extended for the RFTAP by enlarging the matrices in which the (partial) solutions are stored to $|A| \times (|D| + |K|)$-matrices. To keep the pseudopolynomial running time of the algorithm, the number of different types must also be fixed.

Algorithm 2: Exact algorithm

1.Determine Upper Bound:

Run algorithm 1 to determine a heuristic solution X_{tabu} and set $UB = f(X_{tabu})$ and $S^0 = \{\mathbf{0}\}$.

2.Assignment Loop:

for $t = 1, 2, \ldots, n$ **do**
  Set $S^t = \emptyset$.
  for $X = (x_{a,d}) \in S^{t-1}$ **do**
    for $\overline{a} \in A$ **do**
      Create new matrix X' with

$$x'_{a,d} = \begin{cases} x_{a,d} & \text{if } a \neq \overline{a} \\ x_{a,d} + p_{t,d} & \text{if } a = \overline{a} \end{cases}$$

      if $\sum_{a \in A} \sum_{d \in D} w_d \cdot (x'_{a,d} - u_{a,d}) < UB/2$ **then**
        $S^t = S^t \cup \{X'\}$
      end if
    end for
  end for
end for

return $\text{argmin}\{f(X) | X \in S^n\}$

6 Empirical study

For the empirical study, we consider a real-world setting from the automotive industry and some larger, self-generated test instances. The practical example is the following. A car manufacturer in North America employs three hauliers to operate pre-built truck loads of finished vehicles to be moved from factory to dealerships throughout the target market. Transport contracts prescribe that the number of trips as well as the total distance to travel should be distributed according to fixed quotes to be met on a cumulative basis. Apart from that, the number of stops on the trip at different dealerships is an important cost driver for the hauliers. It is currently not considered in the contracts, but since there are ongoing discussions about it, we include it as a third dimension in our study.

We consider the use of real distribution quotes not critical for the meaningfulness of our study results and disguise this sensitive data with made-up quotes. The trips to distribute, however, are coming from the day to day use of the manufacturer's productive planning system.

We look for an equal distribution, so in a perfect partition, every haulier would get exactly one third of the trips, one third of the overall travel distance, and one third of the number of stops. The travel distances range from 11 to 253 kilometers and the number of stops from 1 to 4. The cases we will consider consist of up to 40 trips. The weights are chosen as $w_d = \frac{1}{\sum_{t \in T} p_{t,d}} \; \forall d \in D$. This corresponds to the case that properties and targets are given as relative values and the weight is equal to one in every dimension.

All tests were performed on an Intel Core(TM) i7-3740QM running at 2.70GHz and 16GB of RAM. We determine exact solutions with IBM ILOG CPLEX 12.6.1 and compare them with heuristic solutions obtained from the Tabu Search algorithm. After $l = 50$ iterations without an improvement, the *intensification_phase* is started, which in turn stops after 30 iterations. The *tabu_length* is set to $\lfloor n \cdot m/10 \rfloor$, while it is reduced to $\lfloor n \cdot m/100 \rfloor$ in the *intensification_phase*. The Tabu Search terminates after $r = 10$ restarts or after a total number of $i = 1000$ iterations not including the *intensification_phase*.

It was possible to solve all 36 real-world instances exactly by CPLEX as well as by the Tabu Search. Both approaches were able to solve all instances in less than one second. On average, CPLEX required 0.1 seconds, while the Tabu Search terminated after 0.05 seconds.

Additionally, we generated test instances with 75, 100, and 250 tasks which must be distributed to 5 and 8 agents. Again, we look for an equal distribution, so in every dimension the target values are set to 20% and 12.5%, respectively. For each of these six cases, we examine 20 different instances. As in the real-world-instances, we minimize the relative deviation, so the weights are chosen as $w_d = \frac{1}{\sum_{t \in T} p_{t,d}} \; \forall d \in D$.

We solved all instances by CPLEX with a time limit of one hour and applied the Tabu Search presented in Section 5. All parameters of the Tabu Search coincide with the parameters used to solve the real-world instances. In Table 1 we show for each case how many of the 20 instances were solved exactly by CPLEX and how often the solution of the Tabu Search was at least as good as the solution of CPLEX. Furthermore, we state the average runtimes in seconds. To investigate the robustness of the Tabu Search, we also display the maximum relative deviation of the Tabu Search from the solution of CPLEX given in percent.

Case	# CPLEX opt	∅ runtime CPLEX	# TS ≤ CP	∅ runtime TS	max $\frac{TS-CPLEX}{CPLEX}$ in %
75-5	20	293.5	20	1.4	0
100-5	20	26.8	20	2.6	0
250-5	20	4.8	20	26.7	0
75-8	17	1558.2	5	2.1	0.47
100-8	15	1250.7	18	4.8	0.12
250-8	13	2043.9	20	40.5	0

Table 1: CPLEX vs. Tabu Search for different test cases

Both the Tabu Search and CPLEX solve all instances with 5 agents exactly. It is interesting to observe that the running time of CPLEX decreases when the number of tasks increases. More detailed investigations show that all 20 instances with 250 tasks and 5 agents enable an assignment of the tasks such that the deviation is minimized in every dimension itself. This might be the reason that CPLEX, once the best solution is found, can prove its optimality very fast. On the other hand, the running time of the Tabu Search increases for larger instances as was to be expected.

Regarding the cases with 8 agents, the advantage of the Tabu Search with respect to the runtime becomes obvious. For example, the average runtime of CPLEX for the instances with 250 agents is more than 34 minutes, while the Tabu Search requires just 40.5 seconds on average. Beyond that, the solution of the Tabu Search is at least as good as the solution of CPLEX in all 20 instances. At the expense of higher runtimes, CPLEX leads to better solutions in 15 of 20 instances with 75 tasks and 8 agents. However, even the largest relative gap between the solution of Tabu Search and CPLEX of these 20 instances is less than 0.5%. Note that we used the original parameter setting of the Tabu Search to allow for a fair comparison. Adjusting the parameters to a respective data set certainly allows for further improvements. E.g., if we increase the number of restarts to 30 and the maximum number of iterations to 3000 for dataset 75-8, we obtain 11 optimal instances instead of 5. In conclusion, the Tabu Search is a promising approach to receive solutions of high quality very quickly.

7 Conclusion

Inspired by a real-world problem, we formulated the Fair Task Allocation Problem (FTAP). It deals with the task of finding a fair distribution of jobs to different agents according to specified target values. We presented a mixed-integer model for the problem and identified some properties. Furthermore, we concluded equivalent scheduling formulations for some special cases. In section 4, we extensively analyzed the complexity of the FTAP. Thereby, we figured out that we face an NP-hard problem even if the number of agents and the number of dimensions are fixed. Then we presented two algorithms to solve the FTAP, a Tabu Search procedure and an exact algorithm. The exact algorithm is of theoretical interest because it has pseudopolynomial running time if the number of agents and dimensions are fixed. In our computational study, the Tabu Search obtained very promising results for both the real-world and the generated instances. Especially for cases with many agents, this solution approach seems to be appropriate as CPLEXrequires a lot of time to solve the problem to optimality. In several instances, CPLEX even failed to find the best solution within a time limit of one hour. To achieve a better adaption to problems occurring in practice, the study of an online version of the FTAP might be promising. Furthermore, other concepts for the measure of fairness could be regarded. The formulation of the problem is a very general one, so it could be applied to various other problem areas.

Bibliography

Cattrysse, D. G. and Van Wassenhove, L. N. (1992). A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research*, 60(3):260–272.

Chen, X., Sterna, M., Han, X., and Blazewicz, J. (2016). Scheduling on parallel identical machines with late work criterion: Offline and online cases. *Journal of Scheduling*, 19(6):729–736.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, New York.

Gendreau, M. and Potvin, J.-Y. (2014). Tabu search. In Burke, E. and Kendall, G., editors, *Search Methodologies*, pages 243–263. Springer, Boston, MA.

Glover, F. (1989). Tabu Search - Part I. *ORSA Journal on Computing*, 1(3):190–206.

Graham, R. L., Lawler, E. L., Lenstra, J. K., and Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5:287–326.

Huynh Tuong, N., Soukhal, A., and Billaut, J.-C. (2009). A new dynamic programming formulation for scheduling independent tasks with common due date on parallel machines. *European Journal of Operational Research*, 202(3):646–653.

Jaehn, F. (2016). Sustainable operations. *European Journal of Operational Research*, 253(2):243–264.

Józefowska, J. (2012). Just-in-time scheduling in modern mass production environment. In Ríos-Mercado, R. and Ríos-Solís, Y., editors, *Just-in-Time Systems*, volume 60, pages 171–190. Springer, New York.

Karp, R. (1972). Reducibility among combinatorial problems. In Miller, R. and Thatcher, J., editors, *Complexity of Computer Computations*, pages 85–103. Springer, Boston.

Karsu, Ö. and Azizoğlu, M. (2012). The multi-resource agent bottleneck generalised assignment problem. *International Journal of Production Research*, 50(2):309–324.

Korf, R. E. (2009). Multi way number partitioning. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, pages 538–543.

Korf, R. E. (2010). Objective functions for multi-way number partitioning. In *Proceedings of the Third Annual Symposium on Combinatorial Search*, pages 71–72.

Kovalyov, M. Y. and Werner, F. (2002). Approximation schemes for scheduling jobs with common due date on parallel machines to minimize total tardiness. *Journal of Heuristics*, 8(4):415–428.

Kubiak, W. (2009). *Proportional Optimization and Fairness*, volume 127. Springer Science & Business Media, New York.

Kubiak, W. and Sethi, S. (1991). A note on "level schedule for mixed-model assembly lines in just-in-time systems". *Management Science*, 37(1):121–122.

Kubiak, W., Steiner, G., and Yeomans, J. S. (1997). Optimal level schedule for mixed-model, multi-level just-in-time assembly systems. *Annals of Operations Research*, 69:241–259.

Kuhn, H. (1955). The hungarian method for the assignment problem. *Naval Research Logistics*, 2(1&2):83–97.

Martello, S. and Toth, P. (1995a). The bottleneck generalized assignment problem. *European Journal of Operational Research*, 83(3):621–638.

Martello, S. and Toth, P. (1995b). A note on exact algorithms for the bottleneck generalized assignment problem. *European Journal of Operational Research*, 83(3):711–712.

Mazzola, J. B. and Neebe, A. W. (1988). Bottleneck generalized assignment problems. *Engineering Costs and Production Economics*, 14(1):61–65.

Mazzola, J. B. and Neebe, A. W. (1993). An algorithm for the bottleneck generalized assignment problem. *Computers & Operations Research*, 20(4):355–362.

Miltenburg, J. and Sinnamon, G. (1989). Scheduling mixed-model, multilevel assembly lines in just-in-time production systems. *International Journal of Production Research*, 27(9):1487–1509.

Parrello, B. D., Kabat, W. C., and Wos, L. (1986). Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem. *Journal of Automated Reasoning*, 2(1):1–42.

Pentico, D. W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176(2):774–793.

Sterna, M. (2011). A survey of scheduling problems with late work criteria. *Omega*, 39(2):120–129.

Zhang, J., Mouratidis, K., and Pang, H. H. (2011). Heuristic algorithms for balanced multi-way number partitioning. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 693–698.

V Fazit und Ausblick

Gegenstand dieser kumulativen Dissertation waren zunächst zwei Beiträge aus dem Gebiet der dynamischen Tourenplanung. Während in Beitrag B1 die Modellierung und Lösung einer praktischen Problemstellung aus der Distributionslogistik von Automobilunternehmen im Vordergrund stand, untersuchte Beitrag B2 das Worst-Case-Verhalten von Online-Algorithmen für ein dynamisches Tourenplanungsproblem. Beirag B3 behandelte die Problematik, bereits gebildete Touren für Autotransporte auf mehrere Logistikdienstleister zu verteilen, so dass vertraglich fixierte Quoten möglichst genau eingehalten werden.

Das zentrale Ziel von Beitrag B1 war zu untersuchen, ob sich durch Berücksichtigung von stochastischen Informationen über zukünftige Aufträge Kosteneinsparungen in der Auslieferung fertiger Fahrzeuge an Autohändler realisieren lassen. Dafür wurde eine Lösungsheuristik präsentiert, welche die tägliche Auftragsauswahl anhand der räumlichen Lage der Kunden sowie deren stochastischer Informationen trifft. In einer Fallstudie mit 20 Instanzen à 30 Zeitperioden konnten im Vergleich zu Strategien, welche das Vorgehen in der Praxis widerspiegeln, signifikante Verbesserungen erzielt werden. Im Durchschnitt über alle Instanzen betrugen die Einsparungen gegenüber der besten Vergleichsstrategie 1,85%. Eine weitere interessante Erkenntnis ist, dass die Anzahl an Aufträgen pro Tag einen deutlichen Einfluss auf die erzielten Einsparungen hat. Während bei einer Halbierung der Wahrscheinlichkeiten für Auftragseingänge fast 3% Kostensenkungen möglich sind, liegen diese bei um 50% gesteigerten Auftragswahrscheinlichkeiten nur noch bei gut 0.5%. Somit lässt sich der Schluss ziehen, dass im Rahmen der Auslieferung von Fahrzeugen an Händler Einsparpotenzial durch die Integration stochastischer Informationen besteht, die Höhe allerdings stark von den Instanzen abhängig ist.

Ein alternativer Lösungsansatz besteht in der Formulierung des Problems als stochastisches, dynamisches Programm und der Anwendung von Methoden aus dem Bereich der approximativen dynamischen Programmierung. Einen Einblick in dieses Themengebiet liefert beispielsweise Powell (2011). Unberücksichtigt blieb in diesem Beitrag, inwiefern Fehler in der Prognose der Auftragswahrscheinlichkeiten Einfluss auf die Qualität der Lösungen haben. Solche Tests zur Robustheit des vorgestellten Verfahrens wären daher ein Ansatzpunkt für zukünftige Forschungsarbeiten. Außerdem könnten Saisonalitäten und wöchentlich wiederkehrende Muster im Bestellverhalten der Händler die vorausschauende Tourenplanung beeinflussen. Die Ermittlung solcher Effekte stellt dabei in der Praxis eine Herausforderung für sich dar.

Motiviert durch die praktische Problemstellung im ersten Beitrag wurde in Beitrag B2 ein dynamisches Tourenplanungsproblem untersucht, welches die restriktive Bedingung beinhaltet, dass nur zwei Kunden pro Tour bedient werden können. Abgesehen von Kunden, die in der ersten bzw. letzten Periode bedient werden müssen, stehen zur Belieferung immer zwei aufeinanderfolgende Perioden zur Verfügung. Die Schwierigkeit besteht bei diesem Problem weniger in der Planung der Touren – diese kann mit Hilfe des Algorithmus von Edmonds (1965) vorgenommen werden – sondern vielmehr in der Entscheidung darüber, welche Kunden in einer Periode bedient und welche Kunden auf die nächste Periode aufgeschoben werden sollen. Dabei hat sich gezeigt, dass bereits bei der Beschränkung auf einen Planungshorizont von zwei Perioden kein Algorithmus eine *competitive ratio* unter $\sqrt{2}$ haben kann. Diese untere Schranke steigt bei einer Ausweitung des Planungszeitraums auf mehr als zwei Perioden auf mindestens 1,44 an. Außerdem wurde gezeigt, dass mit dem SMART(p)-Algorithmus ein Verfahren existiert, welches für $p = 2$ bei zwei Perioden Touren erzeugt, deren Gesamtlänge die optimale Gesamtlänge maximal um den Faktor 1,5 übersteigt. Bereits bei drei Perioden ist die *competitive ratio* unabhängig von p jedoch mindestens 1,6. Mit dem Match(λ)-Algorithmus wurde eine Vorgehensweise präsentiert, welche im Fall von zwei Perioden zumindest für eine Teilmenge von

Instanzen eine maximale Abweichung um den Faktor $\sqrt{2}$ garantiert. Die Gültigkeit dieser *competitive ratio* für beliebige Instanzen verbleibt als offene Frage.

Die gefundenen unteren Schranken der *competitive ratio* geben einen interessanten Einblick darin, wie weit die Lösungen von Online-Algorithmen für dynamische Routing-Probleme von der optimalen Lösung abweichen können. Ähnliche Analysen für verwandte Problemstellungen könnten Gegenstand weiterer Forschungsarbeiten auf diesem Gebiet sein. Eine bedeutende Forschungsrichtung im Bereich der dynamischen Tourenplanung ist die Berücksichtigung von stochastischen Informationen, wie Pillac et al. (2013) feststellen und es beispielsweise in Beitrag B1 dieser Dissertation der Fall war. Neben dem Aufzeigen von Einsparpotenzial für praktische Problemstellungen ist auch die theoretische Analyse der Erwartungswerte von Lösungsverfahren interessant.

Der Beitrag B3 hatte die faire Verteilung von Transportrouten auf mehrere Logistikdienstleister zum Thema. Das Ziel war dabei, die zuvor gebildeten Touren so zu verteilen, dass vertraglich fixierte Quoten möglichst genau eingehalten werden. Diese praktische Problemstellung wurde durch ein generisches gemischt-ganzzahliges Programm modelliert, welches auf viele weitere Problemfelder anwendbar ist. Es wurde nachgewiesen, dass es sich um ein NP-schweres Problem im strengen Sinne handelt, so dass zur Lösung des Problems eine Tabu-Suche vorgestellt wurde. Diese konnte alle getesteten, realen Instanzen exakt lösen und fand auch in 103 von 120 generierten, größeren Instanzen eine Lösung, die mindestens so gut war wie die Lösung von IBM ILOG CPLEX mit einem Zeitlimit von einer Stunde. Mit Blick auf die deutlich geringere Laufzeit der Tabu-Suche lässt sich konstatieren, dass diese gut für den Einsatz in der Praxis geeignet ist. Die ebenfalls präsentierte dynamische Programmierung hat gezeigt, dass das Problem bei begrenzter Anzahl an Anbietern und Dimensionen in pseudopolynomieller Zeit lösbar ist. Außerdem wurde nachgewiesen, dass die Problemstellung bei Beschränkung auf eine Dimension äquivalent zu Problemen aus dem Bereich der Ablaufplanung ist, welche die Minimierung von Verspätungen zum Ziel haben.

Die betrachtete Problemstellung kann zu den mehrkriteriellen Optimierungsproblemen gezählt werden, da die Minimierung der Abweichungen von den Zielvorgaben in jeder Dimension als eigenes Ziel angesehen werden kann. Während in diesem Beitrag durch die Zielgewichtung ein einkriterielles Problem gebildet wurde, sind auch andere Ansätze – beispielsweise eine lexikographische Ordnung der Ziele – denkbar. Die Betrachtung einer dynamischen Variante des Problems bietet weiteres Forschungspotenzial. In Anwendungen, in denen die Zuteilung nicht gesammelt erfolgt, sondern unmittelbar bei Eintreffen eines Auftrags, könnten Strategien gefragt sein, welche die Minimierung von Abweichungen über die Zeit hinweg zum Ziel haben. Nicht betrachtet wurden in diesem Beitrag mögliche Kapazitätsengpässe einzelner Anbieter, welche durch eine Erweiterung des aufgestellten Modells Berücksichtigung finden können.

Eine Herausforderung der Zukunft im Bereich der Transportplanung wird der Umgang mit großen zur Verfügung stehenden Datenmengen sein. Die Gewinnung zuverlässiger Informationen aus diesen Daten sowie die Kombination mehrerer stochastischer Aspekte bieten ein weites Forschungsfeld, wie Ritzinger et al. (2016) feststellen. Beispielsweise können Informationen über den Verkehrsfluss in die in dieser Dissertation betrachteten Problemstellungen eingebunden werden. Dies kann einerseits zu weiteren Kosteneinsparungen in der Distribution führen und andererseits die Zufriedenheit der beteiligten Logistikdienstleister erhöhen.

Literaturverzeichnis

Edmonds, J. (1965). Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards*, 69(1-2):125–130.

Pillac, V., Gendreau, M., Guéret, C., und Medaglia, A. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11.

Powell, W. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley, 2. Auflage.

Ritzinger, U., Puchinger, J., und Hartl, R. F. (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231.