Photogrammetrie • Fernerkundung • Geoinformation 5/2006, S. 387–396, 11 Abb., 2 Tab.

Tourenoptimierung mit weichen Zeitfenstern

EIKE LARS MEYER, Hannover & STEFAN SCHÖF, Oldenburg

Keywords: Geoinformatics, traveling salesperson problem, soft time windows, route optimization, tabu search

Summary: *Tour optimization with soft time windows.* This paper presents an algorithm to find an optimal tour where a single vehicle is required to visit each of a given set of stations. This problem is in its basic form well-known as the "Traveling Salesperson Problem". Opposed to many other approaches, our algorithm handles two additional requirements, which are very important to practical applications: The total costs of a tour are defined as a weighted sum of distance and traveling time. Moreover, the vehicle must visit each station within a specified time window. Lateness at stations is allowed but a penalty is added to the costs. The total costs of a tour may be lower if small delays at a station (with small penalties) are accepted. Our algorithm can handle these "soft time windows", too. A heuristic algorithm based on reactive tabu-search is presented. With the help of its implementation it was possible to improve the best-known solutions to some problem instances, which were examined very thoroughly in related papers, before.

Zusammenfassung: In dieser Arbeit wird die für viele Branchen (z. B. Versandhandel) wichtige Frage untersucht, wie ein Fahrzeug eine gegebene Anzahl von Stationen so kostengünstig wie möglich aufsuchen kann. Dieses „Traveling Salesperson Problem“ wurde in der Literatur bereits ausführlich behandelt. Der vorliegende Beitrag berücksichtigt zwei für die Praxis äußerst wichtige Anforderungen und unterscheidet sich dadurch von vielen anderen Arbeiten in diesem Bereich: Die Definition der Kosten einer Route erfolgt über eine gewichtete Summe aus Entfernung und Reisedauer. Außerdem können für die Ankunft an einzelnen Stationen Zeitfenster vereinbart werden, wobei bei Verspätungen Strafzahlungen anfallen können. Die Gesamtkosten einer Route können günstiger sein, wenn man an einer Station geringe Verspätungen (bei entsprechenden geringen Strafzahlungen) in Kauf nimmt. Diese „weichen“ Zeitfenster werden ebenfalls berücksichtigt. Es wird ein heuristischer Algorithmus auf Grundlage der reaktiven Tabusuche vorgestellt. Mittels einer Implementierung des Algorithmus konnten für einige in der Literatur bereits ausführlich untersuchte Probleminstanzen die bisher besten bekannten Lösungen verbessert werden.

1 Einführung

Das Traveling Salesperson Problem (TSP, Problem des Handlungsreisenden) zählt zu den bekanntesten kombinatorischen Optimierungsproblemen. Ein Handlungsreisender unternimmt ausgehend von einer beliebigen Startstation eine Rundreise durch n andere, voneinander verschiedene Stationen, die er genau einmal besucht. Anschließend kehrt er wieder zu seiner Ausgangsstation zurück. Aus den insgesamt $n!$ möglichen Touren soll nun die kürzeste bzw. kostengünstigste Rundreise ermittelt werden, wobei die Entfernungen der einzelnen Stationen zueinander bekannt sind. Bei einer graphentheoretischen Modellierung des Problems werden die Stationen und Wege zwischen diesen als Knoten bzw. Kanten eines vollständigen, gewichteten Graphen definiert. Das Kantengewicht entspricht der Länge bzw. den Kosten des jeweiligen Weges.

Verschiedene Varianten des TSP spielen in der Praxis eine Rolle, so dass manchmal

1432-8364/06/2006/0387 $ 2.50

von einer TSP-Familie (Schmitting 1999) gesprochen wird. So ist es etwa möglich, für Hin- und Rückweg unterschiedliche Kosten anzusetzen (asymmetric TSP) oder bestimmte zeitliche Restriktionen für den Besuch einzelner Stationen zu vergeben (TSP with time windows). Eine wichtige Erweiterung von TSP stellt das Vehicle Routing Problem (VRP, Problem der Tourenplanung) dar, bei dem mehrere Fahrzeuge zum Besuch der Stationen zugelassen sind und zusätzliche Bestellmengen für jede Station beachtet werden. Auch hier gibt es aufgrund verschiedener Praxisanforderungen mehrere Varianten. Es hat sich gezeigt, dass sich Algorithmen zur Behandlung einer bestimmten TSP- oder VRP-Variante nicht ohne weiteres auf andere Varianten übertragen lassen.

Aus den Kundenanforderungen der Firma klickTel, Essen, resultierte die Notwendigkeit, die Variante A-TSP-STW (asymmetric TSP with soft time windows) zu untersuchen, die sich von der oben beschriebenen Grundform wie folgt unterscheidet. Start- und Zielstation können unterschiedlich sein, Hin- und Rückwege können unterschiedliche Kosten haben. Das Fahrzeug muss jede Station innerhalb eines bestimmten Zeitfensters erreichen. Verspätungen hierbei sind erlaubt (weiche Zeitfenster), werden aber mit einer Strafe (Zusatzkosten) geahndet. Optimierungsziel ist eine gewichtete Summe aus Fahrtzeit, Fahrstrecke und Strafkosten. Eine Tour, die Verspätungen an gewissen Stationen und damit Strafzahlungen in Kauf nimmt, kann insgesamt daher trotzdem die kostengünstigste sein.

Eine optimale Lösung für A-TSP bzw. TSP-TW zu finden ist NP-hart (Ascheuer et al. 1997, Calvo 2000, Barnes & Carlton 1996), es können algorithmisch also nur Näherungslösungen in angemessener Zeit ermittelt werden. An der Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven wurde in einer Diplomarbeit (Meyer 2005) eine Heuristik zur Berechnung möglichst kostengünstiger Lösungen entworfen und implementiert. Im Abschnitt 2 dieses Artikels wird zunächst ein kurzer Überblick über verwandte Arbeiten auf dem Gebiet der TSP- und VRP-Algorithmen gegeben. Danach wird die entwickelte Optimierungsheuristik vorgestellt, bevor im nächsten Abschnitt einige Testergebnisse die Leistungsfähigkeit des Verfahrens belegen. Eine kurze Zusammenfassung erfolgt in Abschnitt 5.

2 Verwandte Arbeiten

Im Zusammenhang mit Zeitfenstern findet man in der Literatur wesentlich mehr Arbeiten zum VRP als zum TSP. Das TSP mit weichen Zeitfenstern und asymmetrischen Modellen (Hinweg- und Rückweg haben unterschiedliche Kosten) wurde bislang in der Literatur noch nicht direkt behandelt. Anregungen für einen Algorithmus für das A-TSP-STW sind daher in Arbeiten zum TSP mit Zeitfenstern zu suchen.

Eine erste Formulierung von TSP-TW findet sich in Baker 1983. Hier wird allerdings wie auch in Dumas et al. 1995, Bianco et al. 1997, Gendreau et al. 1998a, Ascheuer et al. 1999, Balas & Simonetti 2001, Focacci et al. 2002 ein exaktes Verfahren zur Bestimmung der optimalen Lösung verwendet. Die Schwierigkeit solcher exakten Verfahren ist jedoch, dass ihre Laufzeit für große Probleme i.a. inakzeptabel ist. Außerdem berichten fast alle Arbeiten, dass die Lösungsschwierigkeit von TSP-TW in erster Linie durch die Größe der Zeitfenster bestimmt wird. Diese können aber bei praktischen Aufgabenstellungen recht groß werden. Ein weiteres Problem ist, dass als Optimierungsziel vorrangig die reine Fahrtzeitminimierung angestrebt wird. Bereits die Berücksichtigung auch der Wartezeiten von Fahrzeugen bei zu frühem Eintreffen bei einzelnen Stationen steigert den Aufwand erheblich (Dumas et al. 1995). Eine Minimierung der tatsächlichen Kosten (d. h. auch Fahrtstrecke und Strafzahlungen werden berücksichtigt) verkompliziert das Problem weiter. Heuristiken, die eine möglichst gute Lösung in annehmbarer Zeit anstreben, erscheinen daher für das zu untersuchende Problem interessanter. Obwohl zu TSP-TW auch Heuristiken mit anderen Strategien entwickelt wurden (z. B. (Gendreau et al. 1998b, Calvo 2000)), zeigt das Verfahren

Abb. 1: Verletzung der Dreiecksungleichung.

aus Barnes & Carlton 1996, das Tabusuche verwendet, sehr gute Ergebnisse: die (sehr schnell) berechneten Lösungen sind maximal 0,8 % von der besten bekannten Lösung entfernt. Auch viele erfolgreiche Heuristiken für das VRP-TW basieren auf Tabusuche, in Taillard et al. 1995 werden sogar weiche Zeitfenster berücksichtigt.

Die Tabusuche hat außerdem den Vorteil, deterministisch zu sein, d. h. bei mehreren Optimierungsläufen für das gleiche Problem wird auch das gleiche Ergebnis erzeugt. Unterschiedliche Resultate könnten beim Anwender hingegen zu Unsicherheiten führen.

Ein weiterer Vorteil der Tabusuche ist, dass sie (im Gegensatz zu vielen anderen Verfahren) die Gültigkeit der Dreiecksungleichung nicht zwingend voraussetzt. Diese fordert, dass der direkte Weg zwischen zwei Punkten nie teurer ist als der Weg über eine Zwischenstation. Obwohl dies in der euklidschen Geometrie immer erfüllt ist, kann es für praktische Tourenplanungsprobleme sinnvoll sein, wegen bestimmter Nebenbedingungen (Vermeidung von Mautstrecken, Bevorzugung von Hauptstraßen etc.) eine Verletzung der Dreiecksungleichung zuzulassen. In Abb. 1 ist zwar der Weg von Station 17 über 16 nach 18 kostengünstiger als der direkte Weg von 17 nach 18, trotzdem soll eine Routenplanung den direkten Weg (Hauptstraße) von 17 nach 18 berücksichtigen. Das im Folgenden vorgestellte Verfahren basiert daher auf der Tabusuche.

3 Der Optimierungsalgorithmus

In diesem Abschnitt wird ein Verfahren für A-TSP-STW vorgestellt. Es stellt eine Adaption des in Barnes & Carlton 1996 vorgestellten Verfahrens dar, das Tabusuche verwendet. Vor einer genaueren Darstellung des Verfahrens werden zunächst die Grundlagen der Tabusuche erläutert.

Einfache Suchheuristiken beginnen mit einer Anfangslösung, die z. B. durch einen Greedy-Algorithmus bestimmt werden kann. Ein Beispiel für einen solchen Algorithmus wäre, ausgehend von der Startstation immer die Station unter den noch nicht zur Route gehörenden hinzuzufügen, die zur zuletzt hinzugefügten Station die geringste Kostendistanz hat. Ausgehend von dieser (in der Regel weit vom Optimum entfernten) Anfangslösung wird nun in jedem Durchlauf von der gerade betrachteten Lösung zu der besten *Nachbarlösung* gewechselt. Ein *Zug* ist dabei die Transformation einer Lösung in eine Nachbarlösung. Die Suche endet, wenn keine Nachbarlösung mehr besser ist als die aktuelle Lösung. In der Regel wird also nur ein lokales, aber nicht unbedingt ein globales Optimum erreicht (siehe Abb. 2).

Was ein gültiger Zug ist (also welche Nachbarlösungen zu einer Lösung existieren), kann für das TSP sehr unterschiedlich definiert werden. Eine Möglichkeit ist es, 2 oder 3 Kanten einer Tour durch neue Kan-

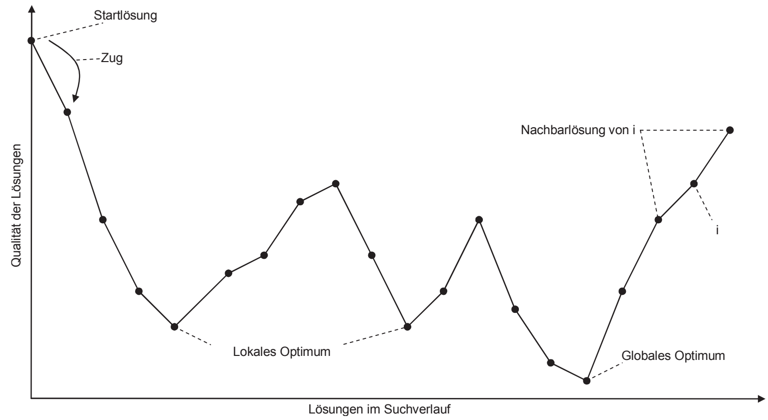

Abb. 2: Nachbarlösungen und Optima.

ten zu ersetzen, um zu einer (eventuell günstigeren) neuen Tour zu gelangen (2-opt- bzw. 3-opt-Verfahren). Eine beliebte Variante zum Erzeugen von Nachbarlösungen ist das Or-opt-Verfahren (Or 1976). Hierbei wird eine Station aus der Tour entfernt und an einer anderen Stelle wieder eingefügt. Dabei werden ebenfalls drei (allerdings nicht beliebige) Kanten durch drei andere ersetzt (siehe Abb. 3). Mit geringerem Aufwand erreicht das Or-opt-Verfahren eine ähnliche Lösungsgüte wie das 3-opt-Verfahren.

Einfache Heuristiken berücksichtigen nur einen relativ kleinen Teilbereich des gesamten Lösungsraumes und bleiben dann in lokalen Optima „stecken“. Die Qualität der gefundenen Lösung hängt damit stark von der Anfangslösung ab und ist ggf. sehr unbefriedigend. Deshalb wurden *Meta-Heuristiken* entwickelt, die diesen Nachteil vermeiden sollen. Zu den bekanntesten Meta-Heuristiken zählen Simulated Annealing,

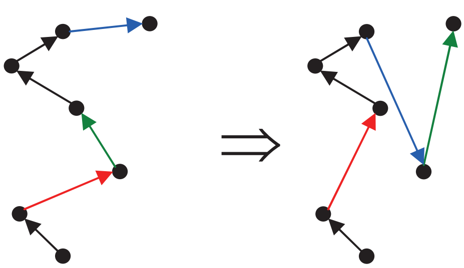

Abb. 3: Or-opt-Verfahren.

genetische Algorithmen und die Tabusuche. Beim *Simulated Annealing* (Kirkpatrick 1983) werden Nachbarlösungen zufällig erzeugt. Ist eine so erzeugte Nachbarlösung besser als die aktuelle, so wird sie als aktuelle Lösung gewählt. Aber auch eine schlechtere Lösung wird mit einer gewissen Wahrscheinlichkeit, die im Laufe der Zeit immer stärker abnimmt, akzeptiert. *Genetische Algorithmen* (Goldberg 1989) benutzen zur Optimierung Prinzipien der biologischen Evolution. Es wird eine Menge von Lösungen erzeugt, wobei durch Kreuzen guter Lösungen neue Lösungen (Nachkommen) erzeugt werden. Eine Lösung kann an einer oder an mehreren Stellen verändert werden (Mutation) und es können besonders gute Lösungen ausgewählt werden (Selektion).

Bei der *Tabusuche* wird jeweils die Nachbarlösung mit den geringsten Kosten als neue aktuelle Lösung ausgewählt, hierbei wird die gesamte Nachbarschaft berücksichtigt. Ein Ersetzen der aktuellen Lösung durch die beste Nachbarlösung findet selbst dann statt, wenn diese Lösung schlechter ist als die aktuelle Lösung. Beim Erreichen eines lokalen Optimums würde auf diese Weise allerdings ständig zwischen diesem lokalen Optimum und dessen besten Nachbarn gewechselt werden. Um dies zu vermeiden, werden die Umkehrzüge der letzten Züge als *tabu* gespeichert und dürfen in den nächsten Schritten nicht mehr ausgeführt werden. Dies ermöglicht das Verlassen lokaler Optima. Die Anzahl der Durchläufe, in denen ein Zug nicht ausgeführt werden darf, wird *Tabulänge* genannt. Eine grundlegende Diskussion der Tabusuche findet sich in Glover & Laguna 1993. Die Qualität der Tabusuche hängt stark von der Tabulänge ab. Ist sie zu klein, werden zyklisch immer wieder dieselben Lösungen gewählt, ist sie zu groß, stehen nicht mehr genügend Züge zur Auswahl, wodurch gute Lösungen evtl. verhindert werden. Bei der *reaktiven Tabusuche* (Battiti & Tecchiolli 1993) wird daher die Tabulänge dynamisch angepasst.

Das in dieser Arbeit vorgestellte Verfahren verwendet die reaktive Tabusuche. Als Anfangslösung wird die Tour gewählt, die sich ergibt, wenn die Stationen in der Reihenfolge ihrer Zeitfenstermittelpunkte sortiert werden. Die in jedem Schritt der Tabusuche zu berücksichtigende Nachbarschaft wird nach dem Or-opt-Verfahren ermittelt. Aus allen so ermittelten Nachbarlösungen wird unter den nicht mit einem Tabu belegten Touren die kostenminimale ausgewählt. Eine Tabuverletzung wird in Kauf genommen, wenn damit eine Lösung erreicht wird, die noch nicht besucht wurde und die kostengünstiger ist als die bisher beste bekannte Lösung. Die Tabusuche endet nach einer festen Anzahl von Iterationen. Alternativ wurden auch zwei Strategien zum vorzeitigen Beenden der Tabusuche implementiert: Beenden nach Mehrfachbesuch der besten Lösung (Taillard et al. 1995) bzw. nach einer Maximalzahl von Schritten ohne Lösungsverbesserung (Osman 1993). Diese Verfahren verkürzten zwar häufig die Laufzeit, wirkten sich dann aber oft negativ auf die Ergebnisqualität aus.

Die Anzahl der Nachbarlösungen kann bei kombinatorischen Optimierungsproblemen sehr groß werden. Um den Algorithmus zu beschleunigen, wurde daher auf zwei Arten versucht, die Anzahl der zu durchsuchenden Nachbarlösungen zusätzlich einzuschränken. Zum einen wurden Nachbarlösungen, in denen Zeitfenster nicht eingehalten werden können, nicht berücksichtigt. Auch eine Beschränkung des Or-opt-Verfahrens auf eine maximale Verschiebedistanz wurde untersucht. Beide Möglichkeiten verschlechterten allerdings die Qualität der Lösungen.

Die Tabusuche erfordert ein schnelles Auffinden und eine effiziente Identifizierung bereits besuchter Lösungen. Um nicht die Reihenfolge jeder besuchten Tour komplett speichern und dann stationsweise vergleichen zu müssen, wird das Hashtourverfahren aus Woodruff & Zemel 1993 verwendet.

Das Berechnen der Kosten einer Tour bestimmt zu einem großen Teil die Laufzeit des Algorithmus. Bei jedem Schritt der Tabusuche muss für jede Nachbartour die Kostenfunktion berechnet werden. Wird nur bzgl. der reinen Fahrzeit minimiert, muss lediglich die Kostendifferenz der Veränderungen von der aktuellen Tour zur Nachbartour berechnet werden. Dies geht verhältnismäßig schnell. Ziel dieser Arbeit ist allerdings die Optimierung der tatsächlichen Kosten als gewichtete Summe aus Fahrt- und Wartezeit, Fahrtstrecke und Strafzahlungen. Hier ist eine komplette Neuberechnung für jede Tour unvermeidbar.

Entscheidenden Einfluss auf die Ergebnisqualität haben die Parameter der reaktiven Tabusuche, vor allem der Tabulängenabnahme- und -zunahmefaktor sowie die initiale Tabulänge. In umfangreichen Testläufen wurde untersucht, wie sich diese Faktoren auf unterschiedliche Probleminstanzen des A-TSP-STW auswirkten. Die Veränderung der initialen Tabulänge wirkt sich in sehr uneinheitlicher Weise auf die Qualität der gefundenen Lösung aus. Ein im Durchschnitt zufrieden stellender Wert wird er-

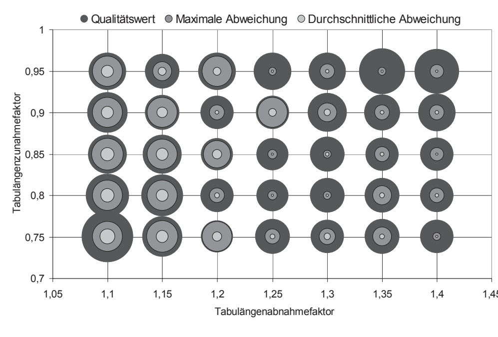

Abb. 4: Qualität der Tabusuche bei verschiedenen Änderungsfaktoren der Tabulänge.

reicht, wenn die Tabulänge anfangs der Anzahl der Stationen entspricht. Bei der Untersuchung der Tabulängenabnahme- und -zunahmefaktoren (siehe Abb. 4) ergab sich, gemittelt über jeweils 8000 Durchläufe von 72 Probleminstanzen für 35 Kombinationen dieser beiden Parameter, ein relativ uneinheitliches Bild. Für die in Abschnitt 4 vorgestellten Tests wurde das Parameterpaar 0,85/1,25 gewählt, das bei guter Durchschnittsqualität nur geringe Abweichungen zeigt. In Barnes & Carlton 1996, Battiti & Tecchiolli 1993 werden mit (0,9/1,2 bzw. 0,9/1,1) ähnliche Werte vorgeschlagen.

4 Testergebnisse

Das im vorigen Abschnitt beschriebene Verfahren wurde in einer .NET-Anwendung unter C# implementiert. Hier können die notwendigen Probleminstanzen eingegeben (siehe Abb. 5) und die Ergebnisse dargestellt werden (siehe Abb. 6 und 7).

Die Laufzeit des Programms nimmt mit steigender Zahl von Stationen zu (siehe Abb. 8). Der Anstieg ist aber verglichen mit einer vollständigen Enumeration (Aufwand in $O(n^n)$) äußerst gering. So erlaubt die reaktive Tabusuche auch das Berechnen von Problemen mit vielen Stationen.

Für eine praxisorientierte Lösung muss nun bestimmt werden, wie viele Iterationen der Tabusuche durchgeführt werden sollen, um ein zufrieden stellendes Ergebnis zu erreichen.

Optimale Lösungen können je nach betrachteter Probleminstanz früh oder auch erst sehr spät gefunden werden. Für das in Abb. 9 dargestellte Problem wird die optimale Lösung bereits nach weniger als 100

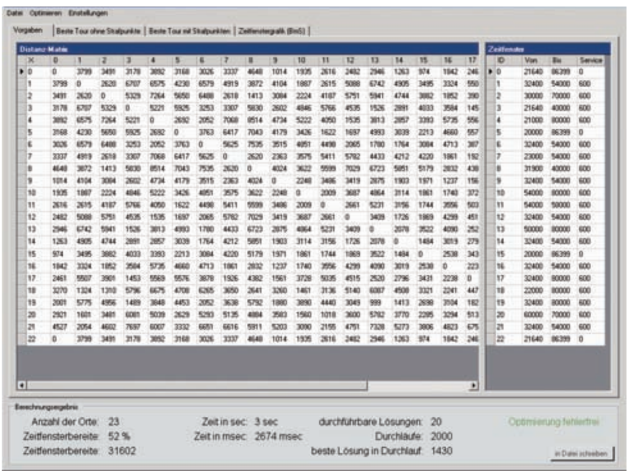

Abb. 5: Hauptfenster der Anwendung.

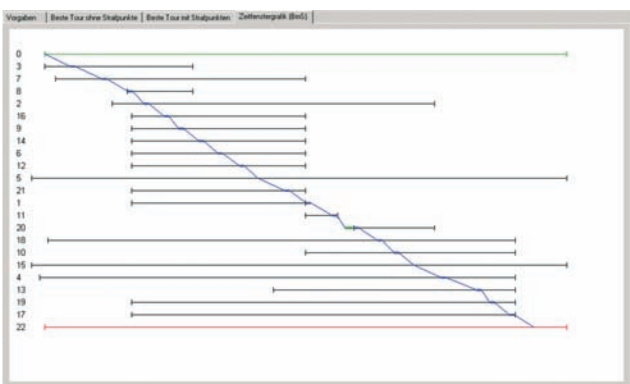

Abb. 6: Zeitfenster-orientierte Darstellung der Ergebnistour.

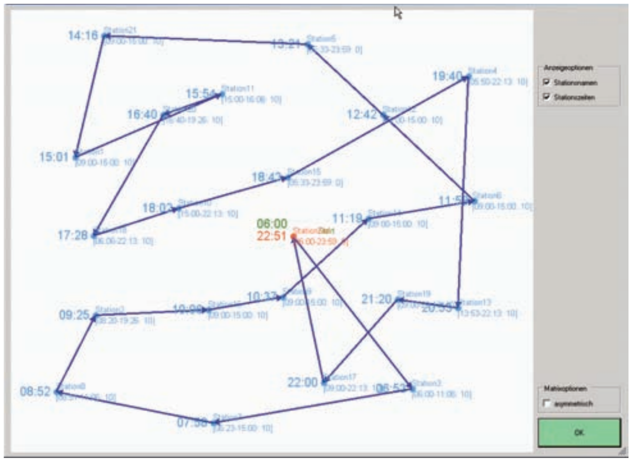

Abb. 7: Räumliche Darstellung der Ergebnistour.

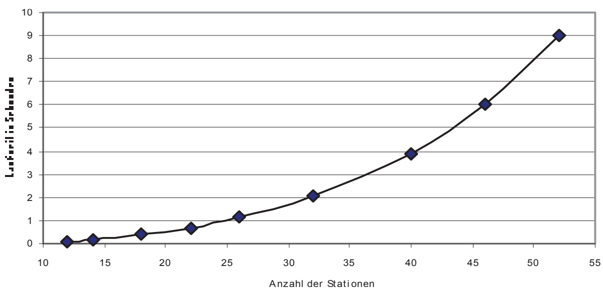

Abb. 8: Laufzeit für 1.000 Iterationen.

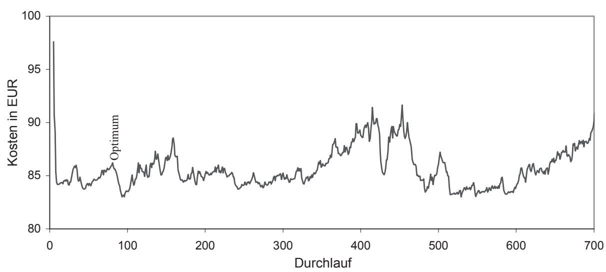

Abb. 9: Kostenverlauf für ein Problem mit 26 Stationen.

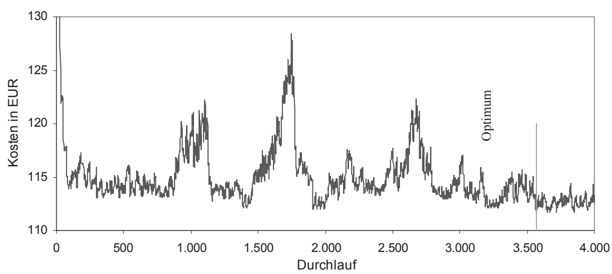

Abb. 10: Kostenverlauf für ein Problem mit 52 Stationen.

Iterationen gefunden, während das in Abb. 10 dargestellte Problem mehr als 3500 Iterationen benötigt. Auch in Abb. 10 ist aber zu sehen, dass schon relativ früh erreichte Lösungen nicht allzu sehr von diesem Optimum abweichen. Selbst die verwendeten Zeitfenster können einen starken Einfluss darauf haben, wann das Optimum gefunden wird. Für das in Abb. 11 dargestellte Routenplanungsproblem wurden je nach gewählten Zeitfenstern zwischen 22 und 2523 notwendige Iterationen gezählt.

Aus einer statistischen Auswertung von 222 Tourenplanungsproblemen aus der Praxis der Firma klickTel wurden daher Richtwerte für die Anzahl der (je nach gewünschter Ergebnisqualität) notwendigen Iterationen abgeleitet (siehe Tab. 1). Beispiel für das Lesen der Tabelle: Damit auf Problemen mit 22 Stationen für 90 % aller Testläufe die ermittelte Lösung um weniger als 1 % von der Bestlösung abweicht, mussten 4415 Iterationen erfolgen. Dies ist also die Mindestzahl empfohlener Iterationen, wenn die gewünschte Ergebnisqualität nur in 10 % aller Fälle verfehlt werden darf.

Um die Qualität des Algorithmus weiter beurteilen zu können, wurde er mit den in Ascheuer 1996 vorgestellten Probleminstanzen getestet. Einige dieser Probleminstanzen konnten mit Branch-and-Cut exakt gelöst werden (Ascheuer et al. 1999). Für die anderen Probleme wurden Schranken angegeben, unterhalb derer sich die optimale Lösung befindet. Durch den in (Focacci et al. 2002) vorgestellten Algorithmus konnten weitere optimale Lösungen angegeben werden. Die Ergebnisse wurden mit Resultaten des in dieser Arbeit präsentierten Algorithmus verglichen. Der hier vorgestellte Algorithmus ist im Gegensatz zu den beiden Vergleichsarbeiten auch für weiche Zeitfenster geeignet und berücksichtigt neben der Distanz auch die Fahrtzeiten. Für die durchgeführten Testläufe wurde eine durchaus mögliche Vereinfachung

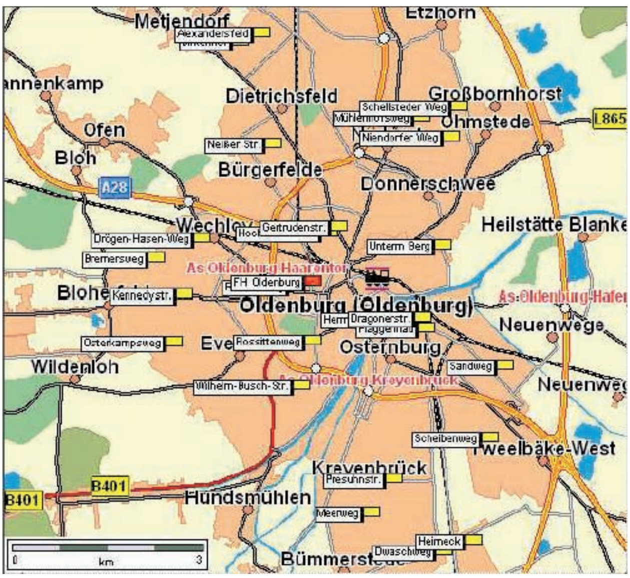

Abb. 11: Routenplanungsproblem mit 27 Stationen.

Tab. 1: Richtwerte für die empfohlene Anzahl von Iterationen.

Anzahl der Stationen	Abweichung = 0 %			Abweichung < 1 %			Abweichung < 5 %		
	zu 97,5 %	zu 90 %	zu 70 %	zu 97,5 %	zu 90 %	zu 70 %	zu 97,5 %	zu 90 %	zu 70 %
14	201	147	87	201	147	87	201	147	87
18	1.553	1.121	639	310	229	139	265	192	110
22	6.295	4.487	2.466	6.225	4.415	2.392	98	76	51
26	5.112	3.744	2.214	1.196	845	453	49	42	33
27	11.642	8.362	4.695	10.545	7.293	3.657	303	221	128
32	24.596	17.845	10.297	6.909	4.895	2.643	1.058	736	377
40	100.514	71.837	39.776	94.309	65.749	33.820	2.780	1.979	1.083
52	352.069	266.603	171.052	221.093	160.691	93.162	59.136	40.876	20.460

Tab. 2: Ergebnisse für die Probleminstanzen aus (ASCHEUER 1996).

Name	# Stationen	[ASCHEUER et al.1999] Wert	Zeit	[FOCACCI et al. 2002] Wert	Zeit	[MEYER 2005] Wert	Zeit	Abw. in %
rbg038a	40	466	4.232,2	466	0,2	466	0,1	0,00
rbg040a	42	386	751,8	386	738,1	386	0,1	0,00
rbg041a	43	382–417	–	403	–	402	511,7	−0,25
rbg042a	44	409–435	–	411	149,8	411	1.946,7	0,00
rbg048a	50	455–527	–	492	–	504	9,6	2,44
rbg049a	51	418–501	–	488	–	484	316,6	−0,82
rbg050a	52	414	18,6	414	95,6	418	66,7	0,97
rbg050b	52	453–542	–	527	–	518	33,2	−1,71
rbg050c	52	509–536	–	543	–	526	209,3	−1,87
rbg055a	57	814	6,4	814	2,5	814	11,2	0,00
rbg067a	69	1048	6,0	1.048	4,0	1.048	102,2	0,00
rbg086a	88	1049–1052	–	1.086	–	1.051	659,2	−0,10
rbg092a	94	1102–1111	–	1.109	–	1.093	65,3	−1,44
rbg125a	127	1410	229,8	1.446	–	1.410	4.357,1	0,00

nicht durchgeführt. In Tab. 2 sind die Testergebnisse für alle Probleme mit mindestens 40 Knoten gezeigt (für kleinere Probleme wurde durch alle Arbeiten die jeweils optimale Lösung erreicht). Wie zu sehen ist, konnte der hier vorgestellte Algorithmus für sechs Probleminstanzen die bisher beste bekannte Lösung verbessern. Darunter ist zum Beispiel ein Problem mit 94 Stationen, für das die bisher beste bekannte Lösung innerhalb von 66 Sekunden um 1,44 % verbessert werden konnte. Insgesamt konnte lediglich für zwei Probleme die (bisherige) Bestlösung nicht erreicht werden.

5 Zusammenfassung

Es wurde ein auf der reaktiven Tabusuche basierendes Verfahren für die Tourenoptimierung mit weichen Zeitfenstern vorgestellt. Alle Tourenplanungssysteme, die mit Zeitfensterrestriktionen arbeiten, können Touren mit undurchführbaren Zeitfenstern nicht optimieren; hier verteilen sie zum Beispiel die Tour auf mehrere Tage. Die Berücksichtigung von undurchführbaren Touren kann trotz Strafzahlungen zu einer insgesamt kostengünstigeren Tour führen. In vielen Testinstanzen konnte beobachtet werden, dass minimale Zeitüberschreitungen einen beachtlichen Kostenvorteil bringen können.

Die Kostenberechnungsfunktion kann variabel eingesetzt werden. So können auch weitere streckenabhängige Kosten (z. B. Maut) in die Kostenoptimierung einfließen.

Der implementierte Algorithmus ermittelt in der Praxis für Touren mit bis zu 30 Stationen das globale Optimum auf einer 3-GHz-CPU innerhalb von 10 Sekunden. Sogar bei Touren mit bis zu 52 Stationen werden sehr gute Lösungen gefunden, die im Durchschnitt nicht weiter als 1 % vom globalen Optimum abweichen. Allerdings muss bei dieser Menge an Stationen im Einzelfall auch mit höheren Abweichungen gerechnet werden. Auch im Vergleich zu anderen Verfahren und Probleminstanzen überzeugt der Suchalgorithmus. So konnten bei mehreren bekannten Problemen die bisherigen Bestmarken verbessert werden.

Eine Erweiterung der Aufgabenstellung besteht in der Berücksichtigung mehrerer Fahrzeuge (M-TSP-STW) und zusätzlicher Bestellmengen (VRP-STW). Eine gute Lösung hierzu muss sich nicht ohne weiteres aus dem hier vorgestellten Algorithmus ergeben. Ein Ansatz auf Basis der reaktiven Tabusuche erscheint aber viel versprechend.

Literatur

ASCHEUER, N., 1996: Hamiltonian Path Problems in the On-line Optimization of Flexible Manufacturing Systems. – Technical Report TR 96-3, Konrad-Zuse-Zentrum.

ASCHEUER, N., FISCHETTI, M. & GRÖTSCHEL, M., 1997: A polyhedral study of the asymmetric travelling salesman problem with time windows. – Preprint SC 97–11, Konrad-Zuse-Zentrum.

Ascheuer, N., Fischetti, M. & Grötschel, M., 1999: Solving the Asymmetric Travelling Salesman Problem with Time Windows by Branch-and-Cut. – Preprint SC 99–31, Konrad-Zuse-Zentrum.

Baker, E.K., 1983: An Exact Algorithm for the Time-Constrained Traveling Salesman Problem. – Operations Research **31(5)**: 983–945.

Balas,E. & Simonetti,N., 2001: Linear Time Dynamic-Programming Algorithms for New Classes of Restricted TSPs: A Computational Study. – INFORMS Journal on Computing **13**(1): 56–75.

Barnes, J.W. & Carlton, W.B., 1996: Solving the traveling-salesman problem with time windows using tabu search. – IIE Transactions **28:** 617–629.

Battiti, R. & Tecchiolli, G., 1994: The reactive tabu search. – ORSA J. Comput., **6**(2): 126–140.

Bianco, L., Mingozzi, A. & Ricciardelli, S., 1997: Dynamic Programming Strategies for the Travelling Salesman Problem with Time Windows and Precedence Constraints. – Operations Research **45** (3): 365–377.

Calvo, R.W., 2000: A New Heuristic for the Traveling Salesman Problem with Time Windows. – Transportation Science **34** (1): 113–124.

Dumas, Y., Desrosiers, J., Gelinas, J. & Solomon, M. M., 1995: An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. – Operations Research **43** (2): 367–371.

Focacci, F., Lodi, A. & Milano, M., 2002: A Hybrid Exact Algorithm for the TSPTW. – Journal on Computing **14:** 403–417.

Gendreau, M., Pesant, G., Potvin, J.Y. & Rousseau, J.M., 1998a: An Exact Constraint Logic Programming Algorithm for the Traveling Salesman Problem with Time Windows. – Transportation Science **32** (1): 12–29.

Gendreau, M., Hertz, A., Laporte, G. & Stan, M., 1998b: A generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows – Operations Research **46** (3): 330–335.

Glover,F. & Laguna, M., 1993: Tabu Search. – In: Reeves, C.R. (Ed.): Modern Heuristic Techniques for Combinatorial Problems. – Blackwell Scientific Publications, 70–150.

Goldberg, D.E., 1989: Genetic Algorithms in Search, Optimization, and Machine Learning. – Addison-Wesley.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M.P., 1983: Optimization by Simulated Annealing. – Science, **220** (4598): 671–680.

Meyer, E.L., 2005: Eine praxisorientierte Lösung der Tourenoptimierung mit Zeitfenstern. – Diplomarbeit, Fachhochschule Oldenburg/Ostfriesland/Wilhelmshaven.

Or, I., 1976: Traveling Salesman-Type Combinatorial Problems and Their Relation to the Logistics of Regional Blood Banking. – Dissertation, Northwestern University, Evanston.

Osman, I.H., 1993: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. – Annals of Operations Research **41:** 421–451.

Schmitting, W.,1999: Das Traveling-Salesman-Problem – Anwendungen und heuristische Nutzung von Voronoi-/Delaunay-Strukturen zur Lösung euklidischer, zweidimensionaler Traveling-Salesman-Probleme. – Dissertation, Universität Düsseldorf.

Taillard, E.D. et al., 1995: A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. – Technical Report CRT-95–66, Université de Montréal.

Woodruff, D.L. & Zemel, E., 1993: Hashing vectors for tabu search. – Annals of Operations Research **41:** 123–137.

Anschriften der Autoren:

Prof. Dr. rer. nat. Stefan Schöf
Institut für Angewandte Photogrammetrie und Geoinformatik, FH Oldenburg/Ostfriesland/Wilhelmshaven
Ofener Straße 16/19, D-26121 Oldenburg
Tel.: +49-441-7708 3323, Fax: +49-441-7708 3336
e-mail: schoef@fh-oldenburg.de

Dipl.-Ing. Eike Lars Meyer
Charlottenstr. 83, D-30449 Hannover
e-mail: mail@elmeyer.de

Manuskript eingereicht: Mai 2006
Angenommen: Juni 2006