FernUniversität

Gesamthochschule in Hagen

Fachbereich Wirtschaftswissenschaft

Lehrstuhl Wirtschaftsinformatik

Seminararbeit zum Thema

Praktische Anwendungen der Suchstrategie Tabu Search - ein Überblick

Seminar:	Wirtschaftsinformatik
bei:	Prof. Dr. H. Gehring
Matr.-Nr.:	4 543 645
Name:	Christian H. Kuhn
Anschrift:	Bentheimstraße 11, 97072 Würzburg
Telefon:	(09 31) 8 04 75 38
Abdabedatum:	27.05.2001
URL:	`http://www.qno.de/wiwi/winf/semref.pdf`

Inhaltsverzeichnis

1 Was ist Tabu Search?

1.1 Problemstellung

In vielen Bereichen des Lebens tauchen Optimierungsprobleme der Art

$$\begin{aligned} \text{Minimiere} \quad & f(x) \\ \text{unter der Einschränkung} \quad & g_i(x) \geq b_i; \quad i = 1, \dots, m; \\ & h_j(x) = c_j; \quad j = 1, \dots, n. \end{aligned} \tag{1.1}$$

auf. x ist hier ein Vektor von Entscheidungsvariablen, f, g_i und h_j allgemeine Funktionen. In den hier betrachteten Fällen können die Entscheidungsvariablen nur diskrete Werte annehmen. Auch bereitet die exakte mathematische Formulierung der Einschränkungen in Form von Gleichungen oder Ungleichungen mitunter Schwierigkeiten, weshalb man auch die Formulierung

$$\begin{aligned} \text{Minimiere} \quad & f(x) \\ \text{unter der Einschränkung} \quad & x \in \mathbf{X} \end{aligned} \tag{1.2}$$

benutzt; dies erlaubt auch verbale Formulierung der Einschränkungen, wo dies angebracht erscheint.

Die Funktion $f(x)$ wird oft nicht nur ein oder mehrere globale Optima aufweisen, sondern auch verschiedene lokale Optima. Zur Beschreibung wird das Konzept der Nachbarschaft verwendet: Eine Nachbarschaft $N(x, \sigma)$ einer Lösung x umfaßt alle Lösungen, die durch *eine* Elementaroperation σ, einen *Zug*, von x aus erreicht werden. Solche Änderungen können die Änderung einer Entscheidungsvariable um einen Schritt, das Hinzufügen oder Entfernen eines Objekts oder das Vertauschen zweier Objekte sein. Wenn eine Lösung x besser ist als alle ihre Nachbarn, d.h. $f(x) < f(y), y \in N(x, \sigma)$, dann ist x ein lokales Optimum bzgl. dieser Nachbarschaft. Benennt man eine zusammenhängende Teilmenge des Lösungsraums eine Region, dann ist das „globale" Optimum dieser Region ein regionales Optimum.

Ein lokales Optimum ist nicht notwendigerweise auch ein globales Optimum. Es gibt Methoden, die das Auffinden des globalen Optimums garantieren. Der Zeitbedarf solcher Methoden steigt jedoch für die hier betrachteten kombinatorischen Probleme exponentiell mit der Anzahl der zu betrachtenden Objekte. Trotz der Zunahme der Rechengeschwindigkeit erschwinglicher Rechner ist der Einsatz solcher Methoden daher nicht immer sinnvoll. Statt dessen gelangen Methoden zum Einsatz, die nicht die beste Lösung — ein globales Minimum — suchen, sondern eine gute, fast optimale Lösung, diese aber mit einem vertretbaren Aufwand. Eine solche Methode nennt man Heuristik. Von den möglichen Heuristiken wird im Folgenden die Tabu Search (TS) betrachtet.

1.2 Beschreibung von TS

Tabu bezeichnet lt. LANGENSCHEIDT-VERLAG (2001)

> das; -s,-s 1. unausgesprochenes gesellschaftliches Verbot, über ein bestimmtes Thema zu sprechen 2. ungeschriebenes Gesetz, gesellschaftliches (moralisches, rel.) Verbot, et. zu tun 3. (rel.) mystisch begründetes Verbot, et. oder jemanden zu berühren (meist bei Naturvölkern).

TS enthält keine moralisch oder religiös begründeten Verbote. Vielmehr wird hier ein Suchprozeß durch Einschränkungen geführt. Suchmöglichkeiten werden durch Tabus verboten, ihre Evaluierung verändert oder ihre Auswahlwahrscheinlichkeit verändert.

1.2.1 Demonstrationsproblem

Das Prinzip von TS wird an einem Permutationsproblem demonstriert: Eine Lösung bestehe aus einer Anordnung von n Elementen, die durch eine Zahl eindeutig gekennzeichnet sind. Der Lösungsvektor x enthält dann die Ordnungszahlen der Elemente in der Reihenfolge ihres Auftretens. Ein Zug vertausche zwei Elemente miteınander. Ein Zug σ wird also durch die vertauschten Elemente (x_1, x_2) gekennzeichnet. Da der Zug (x_1, x_2) dem Zug (x_2, x_1) äquivalent ist, wird i.d.R. das kleinere Element zuerst genannt. Im folgenden wird ohne Beschränkung der Allgemeinheit davon ausgegangen, daß $x_1 < x_2$. Jedem möglichen Vektor x wird ein Wert der Zielfunktion $f(x)$ zugeordnet.

1.2.2 Nachbarschaftsbildung

Ausgehend von einem — zufällig oder durch Schätzung erhaltenen — Ausgangsvektor wird eine Liste der möglichen Züge erstellt. Die durch die Anwendung dieser Züge auf den Ausgangsvektor entstehenden Vektoren bilden die Nachbarschaft des Ausgangsvektors $N(x, \sigma)$. Jedem Zug wird der Zielfunktionswert des entstehenden Vektors zugeordnet und die Liste der möglichen Züge nach dem resultierenden Zielfunktionswert abfallend sortiert. Das bisher erhaltene Optimum mit dem dazugehörigen Vektor wird zwischengespeichert als optimale Lösung. Der an der Spitze der Liste stehende Zug wird ausgeführt, der erhaltene Wert mit der optimalen Lösung verglichen, diese u.U. aktualisiert, und wieder wird eine Liste der möglichen Züge erstellt, sortiert usw. usf., bis ein Abbruchkriterium erreicht wird. Es ist nun denkbar, daß auf diese Weise die immer gleichen zwei oder mehr Zustände zyklisch durchlaufen werden, ohne daß je ein Abbruch erreicht wird.

1.2.3 Tabubildung

Zur Vermeidung des Entstehens solcher Zyklen wird dem ausgeführten Zug (x_1, x_2) ein Tabuwert zugeordnet. Die Tabuwerte aller möglichen Züge werden günstig in einem zweidimensionalen Feld abgelegt; aufgrund der Konvention über die Zugbezeichnung wird hier nur die rechte obere Hälfte benutzt. Ein solcher Tabuwert kann dauerhaft sein und den gekennzeichneten Zug für immer verbieten; in diesem Fall wäre es sinnvoll, die Matrix mit Nullen zu initialisieren und als Tabuwert irgendeinen von Null verschiedenen Wert einzusetzen. Das Tabu kann in seiner Geltungsdauer auch auf eine bestimmte Anzahl von Zügen beschränkt werden; in diesem Falle setzt man den Wert auf diese Anzahl und reduziert nach jedem weiteren ausgeführten Zug jeden Tabuwert $\neq 0$. Steht an der Spitze der in 1.2.2 erstellten Liste ein Zug mit einem Wert $\neq 0$, wird dieser ignoriert (Ausnahme: Aspirationskriterien, siehe

Abschnitt 1.2.5 auf der nächsten Seite).

Eine weitere Möglichkeit besteht in der Tabuisierung eines Zugs nicht oder nicht nur nach seinem Auftreten in der Vergangenheit (*recency*), sondern nach der Häufigkeit (*frequency*) seines Auftretens. Hierzu wird auch die Häufigkeit des Auftretens eines Zuges in der Tabumatrix abgelegt, indem für einen Zug (x_1, x_2) der Eintrag des inversen Zuges (x_2, x_1) bei jedem Auftreten des Zuges um 1 erhöht wird. Dieser Häufigkeitswert wird dann vom Wert der nach möglicher Ausführung des Zuges erhaltenen Zielfunktion f abgezogen. Dies setzt voraus, daß die Unterschiede der Zielfunktionen zweier verschiedener Ergebnisvektoren in der Größenordnung nicht zu großer natürlicher Zahlen liegen, so daß ein häufiges Auftreten eines Zuges während des Suchvorgangs zwar ein Herabsetzen der Zielfunktion bewirkt und damit ähnlich gute Lösungen zum Zuge kommen läßt, nicht aber sehr viel schlechtere.

Unabhängig vom Auftreten eines Zuges können Tabukriterien auch durch Attribute eines Zuges bestimmt werden. Attribute sind Werte, die von den durch einen Zug herbeigeführten Änderungen abhängen, ohne schon durch $f(x)$ oder gar x selbst erfaßt zu sein. So kann eine binäre Variable durch einen Zug ihren Zustand ändern. Mögliche Tabukriterien wären in einem solchen Fall die Änderung der Variablen zurück in den ursprünglichen Zustand oder aber auch die Beibehaltung des Zustandes. Liegen mehrere solcher Variablen vor, sind auch alle logischen Kombinationen als Tabukriterien möglich.

Attributbezogene Tabukriterien werden oft nur angewandt, wenn die entsprechende Attributeigenschaft innerhalb einer beschränkten Zügezahl auftrat (sog. *recency–based restriction*) oder über einen längeren Zeitraum mit einer gewissen Häufigkeit auftraten (sog. *frequency-based restriction*). Abhängig von diesen Regeln wird dem Attribut dann der *Tabu-Status* tabu-aktiv oder tabu-inaktiv zugeteilt.

1.2.4 Abbruchkriterien

Ziel von TS ist es, in endlicher Zeit eine hinreichend gute Lösung zu finden. Während durch Tabus versucht wird, eine zu frühe, schlechte Lösung zu vermeiden, wird durch geeignete Abbruchkriterien entschieden, wann die Suche zu beenden ist. Das einfachste Abbruchkriterium stellt das Erreichen eines lokalen Optimums dar:

$$\bigwedge_{y \in N(x,\sigma)} f(x) < f(y) \tag{1.3}$$

D.h. es kann in der Nachbarschaft von x kein Zustand gefunden werden, der günstiger als x wäre. Problematisch hierbei ist, daß oft lokale Optima auftreten, die keine globalen Optima sind.

Um dieses Problem zu vermeiden, werden Zufallsprozeduren hinzugefügt. So wird in der „Monte-Carlo“-Methode zufällig ein Vektor aus der Nachbarschaft ausgewählt. Führt diese Auswahl zu einer Nicht-Verschlechterung in der Zielfunktion, wird der ausgewählte Zug angenommen und ausgeführt. Führt die Auswahl zu einer Verschlechterung in der Zielfunktion, wird der Zug dennoch mit einer gewissen Wahrscheinlichkeit akzeptiert, die vom Ausmaß der Verschlechterung abhängt und mit zunehmender Differenz der (evtl. um den *frequency*-Wert korrigierten) Zielfunktionswerte abnimmt. Der Abbruch erfolgt, wenn aus einer Situation heraus kein Zug mehr ausgeführt wird, also weder eine Verbesserung möglich ist noch ein verschlechternder Zug zufällig angenommen wurde. Durch Absenken der Annahmewahrscheinlichkeit eines verschlechternden Zuges mit zunehmender Anzahl der insgesamt ausgeführten Züge kann die Suchdauer begrenzt werden.

1.2.5 Aspirationskriterien

Es mag vorkommen, daß ein Zug durch Tabus verboten ist, der bei Ausführung eine große Verbesserung der Zielfunktion bewirken würde. In solchen Fällen mag es sinnvoll sein, das Ausführen eines tabuisierten Zuges doch zu erlauben. Regeln für diese Erlaubnis werden *Aspirationskriterien* genannt, das Ausmaß der Veränderung der Zielfunktion *Einfluß*.

Aspirationskriterien können einen Zug oder ein Attribut betreffen. Im ersten Falle widerrufen sie die Tabuisierung eines Zuges. Im zweiten Fall wird der Tabu-Status des Attributs auf tabu-inaktiv gesetzt, was aber nicht die Enttabuisierung eines Zuges — die ja auch von anderen Tabus abhängen kann — garantiert.

Ein typisches Beispiel für ein Aspirationskriterium ist die Enttabuisierung eines Zuges, durch den ein globales oder regionales Optimum erreicht werden kann, was sofort einsichtig ist. Ebenso klar ist die Aspiration des „am wenigsten“ tabuisierten

Zuges in einer Situation, in der alle möglichen Züge tabu sind. Auch kann es nützlich sein, einen Zug mit geringem Einfluß aspiriert werden, wenn seit ihrer Tabuisierung ein Zug mit hohem Einfluß durchgeführt wurde.

Eine weitere Verfeinerung kann erfolgen, wenn der Tabustatus nicht nur durch verboten/tabu-aktiv und erlaubt/tabu-inaktiv beschrieben wird, sondern auch eine Zwischenstufe „schwebend“. Ein schwebender Zug kann entweder nur zugelassen werden, wenn es keine Lösungen gibt, die die Zielfunktion verbessern, oder — in der strikteren Fassung — wenn er darüber hinaus selbst die Zielfunktion verbessert.

Weitere Einzelheiten können bei GLOVER UND LAGUNA (1993) gefunden werden, dem auch alle Angaben aus Kapitel 1 entnommen sind.

2 Einsatzgebiete

Die Einsatzgebiete für TS sind weit gestreut. GLOVER UND LAGUNA (1993) gibt auf S. 128 einen umfassenden Überblick. Eine Internet-Recherche bei GOOGLE (2001) nach Tabu-Search-Anwendungen unter Ausschluß des Suchworts Nr. 1 ergab 6460 Treffer. Wenn auch durch weitere Einschränkungen die Zahl der Fundstellen reduziert werden konnte, zeigt dieses Ergebnis doch, daß eine umfassende Übersicht im Rahmen einer Seminararbeit nicht möglich ist. Daher kann nur ein ausgewählter Teil dieser Ergebnisse hier zusammengefasst und gegliedert dargestellt werden.

2.1 Informatik/KI

Bei der Erstellung komplexer Software („Software-Engeneering") treten Probleme auf, die durch heuristische Methoden gelöst werden, nachdem die Probleme als Optimierungsprobleme neuformuliert worden sind; unter anderem kommt hier auch TS zum Einsatz. CLARKE U. A. (2000) beschreibt insbesondere den Einsatz von Heuristiken beim Testen von verschiedenen möglichen Entwicklungsvarianten. Hier wird auch gezeigt, wie Probleme des Software Engeneering so formuliert werden können, daß sie der Lösung durch TS zugänglich sind. In SEXTON U. A. (1999) wird TS als überlegene Methode bei der Optimierung künstlicher neuronaler Netzwerke beschrieben.

2.2 Zeitplanung

Quer durch alle Bereiche von Industrie und Geschäftswelt stellt die zeitliche Einsatzplanung von Mitarbeitern und Gerät ein wichtiges Problem dar. So ist es nicht zu verwundern, daß in diesem Bereich viele TS-Anwendungen zu finden sind.

Standardisierte Problemstellung finden sich für Produktionsprobleme: Eine oder mehrere Maschinen, parallele Fertigung, Kosten und Dauern von Bearbeitungsschritten vorgegeben oder durch die Planung beeinflußbar. Als einer grundlegenden Problemstellung wurde dem *„permutation flow shop"*–Problem frühe Aufmerksamkeit zuteil. n Aufträge aus mehreren Arbeitsschritten sollen so auf m ständig verfügbare Maschinen verteilt werden, daß die Herstellungsdauer aller Aufträge minimal wird. WIDMER UND HERTZ (1989) zeigt die Überlegenheit von TS bei dieser Aufgabenstellung gegenüber bisherigen Methoden.

Das *flow shop*–Problem tritt selten in der beschriebenen Form auf. In der Praxis sind Variationen zu finden, für die auch die Lösungen variieren müssen. Ein Beispiel hierfür bieten LAGUNA U. A. (1993), man beschreibt die Optimierung der zeitlichen Reihenfolge verschiedener Arbeiten an einer Maschine unter Beachtung von Rüstzeit, -kosten und Verzögerungskosten von Aufträgen als theoretisches Problem. Das Problem der Minimierung der Herstellungsdauer bei unabhängigen parallelen Maschinen wird in SRIVASTAVA (1998) gelöst.

Auch komplexere Probleme können mit Heuristiken gelöst werden. Die Wartung von Gleisbauten soll einerseits den regulären Zugbetrieb nicht stören, andererseits soll — aus Kosten- und Sicherheitsgründen — die Arbeitszeit minimiert werden. Eine TS-Lösung dieses Problems unter der Berücksichtigung weiterer materieller und zeitlicher Beschränkungen wird in HIGGINS (1998) vorgestellt.

TS-basierte *scheduling*–Algorithmen treten auch abseits der üblichen Anwendungen in Produktion und Verkehr auf: BANERJEA-BRODEUR U. A. (1998) beschreibt den Einsatz von TS in der Logistik einer großen Klinik: Die Planung der Lieferung von Wäsche sowohl hinsichtlich der Menge als auch des Lieferzeitpunkts wird durch einen TS-Algorithmus gelöst.

Die Problematik der gesamten Planung des öffentlichen Personenverkehrs ist — von der Netzoptimierung über die Optimierung des Einsatzes von Fahrzeugen und Personal bis hin zur Kontrolle — in DADUNA U. A. (1993) umfassend von vielen Authoren ausgeführt. Hierauf wird in Kapitel 3 näher eingegangen.

2.3 Wegoptimierung

Das Problem der Wegoptimierung wird auch oft unter der Überschrift „Travelling Salesman“ behandelt. Hier ist eine Route durch n Punkte so zu planen, daß die Fahrzeit, die Strecke oder die Kosten minimal werden. Erste TS-Lösungen wurden von MALEK U. A. (1989); GLOVER (1992) veröffentlicht.

Aktualität hat dieses Problem gewonnen durch die Entwicklung von Routenplanern fürs Internet, die in kurzer Zeit die nach verschiedenen Kriterien optimierte Route zwischen zwei Orten ermitteln müssen. Einen Einblick im Rahmen einer Vorlesung gibt GAMBARDELLA (2000).

Auch hier tritt das Problem nicht immer in der reinen Form auf; die Praxis gibt Variationen vor. Eine Lösung des „multi-trip“-Wegoptimierungsproblems, bei dem ein Fahrzeug im Planungszeitraum mehr als eine Tour fährt, hier für einen Hersteller von Biscuits, bietet BRANDAO UND MERCER (1998) an.

In theoretischen Problemen sind die während einer Route Aufgesuchten vierundzwanzig Stunden am Tag erreichbar. Dies trifft im richtigen Leben so oft nicht zu. Eine Ergänzung des TS-Algorithmus bei der Wegoptimierung unter der Berücksichtigung von Zeitfenstern wird in LIM U. A. (2000) aufgezeigt.

Eine Wegoptimierung ist auch bei der Verlegung von Netzwerkkabeln möglich. Eine Annäherung an ein optimales Glasfaser-Netzwerk inclusive der Auswahl der Standorte von Multiplexern und Übergabepunkten wird in LEE U. A. (2000) gefunden.

3 Tabu Search bei der Synchronisation von Fahrplänen im ÖPV

Die Ausführungen dieses Kapitels folgen DADUNA UND VOSS (1993).

3.1 Aufgabenstellung

3.1.1 Allgemeines

Im Allgemeinen erfolgt die Betrachtung des Öffentlichen Personenverkehrs (ÖPV) unter einem so großen Blickwinkel, daß das betrachtete Bedienungsgebiet nicht nur von einer Route (Linie) erschlossen wird. Um von einem Punkt A zu einem Punkt B zu gelangen, der nicht von derselben Linie bedient wird, ist für den Fahrgast also ein Umsteigen erforderlich. Hierbei entstehen an den Übergangsknotenpunkten Wege (wenn die Haltepunkte der verschiedenen Linien nicht identisch sind, z.B. Ankuft und Abfahrt auf verschiedenen Bahnsteigen, aber auch u.U. Bushaltestellen, die mehrere Häuserblocks auseinanderliegen) und Wartezeiten (zwischen Ankuft des Fahrgastes und Abfahrt der neuen Linie). Weg- und Wartezeiten zusammen werden als Übergangszeit zusammengefaßt.

Die zitierten Authoren versuchen in Ihrem Artikel, eine Optimierung der Fahrplanabstimmung verschiedener Linien unter dem auch heute noch in der Praxis ungewöhnlichen Blickwinkel der Minimierung der Übergangszeit des Fahrgastes zu erreichen.

3.1.2 Annahmen

Die erste Annahme betrifft das Liniennetzwerk: Die Linienführung einer Linie seien bekannt und unveränderlich. Hierbei werden die beiden möglichen Richtungen einer Linie (Hin- und Rückfahrt) im Interesse einer Vereinfachung des Modells als zwei verschiedene Linien angesehen. Die Linien werden durch Start- und Endpunkte sowie die Übergangsknotenpunkte charakterisiert.

Die zweite Annahme betrifft die interne zeitliche Planung *einer* Linie: Die Fahrzeiten zwischen den Knotenpunkten sei bekannt und vorgegeben. Auf eine Optimierung, in die auch Wartezeiten einer Linie an einer Haltestelle (z.B. auf einen übergeordneten Anschluß) einfließen, wird also in diesem Modell verzichtet.

Die dritte Annahme schließlich betrifft die Knotenpunkte: Die Übergangszeiten der Fahrgäste in den Knotenpunkten sowie die Übergangshäufigkeit sei bekannt und konstant. Das Modell läßt also die in der Motivation noch bewußte Auswirkung von Wartezeiten auf die Entscheidung möglicher Fahrgäste in der Konkurrenz zwischen öffentlichem und Individualverkehr außer Acht.

Weitere Einschränkungen werden außerhalb der Modellstruktur formuliert, da ihre explizite mathematische Formulierung einen unverhältnismäßig großen Aufwand bedeuten würde:

- Die Anzahl der Übergänge sei von der Wartezeit unabhängig.
- Mögliche Alternativrouten (von A nach B mit Linie 1, entweder von B nach C mit Linie 2 und von C nach D mit Linie 3 oder von B über E nach D mit Linie 4) werden nicht betrachtet.
- Die bereitgestellte Transportkapazität wird als ausreichend vorausgesetzt; der Einsatz kleinerer oder größerer Fahrzeuge oder längerer Züge wird hier nicht betrachtet.

Diese Annahmen und Einschränkungen gehen also von einer großen Menge als bekannt vorausgesetzter Daten aus, die vor einer Berechnung eines Optimums erhoben werden müssen. Die Qualität der Lösung ist von der Qualität der Datenerhebung abhängig.

3.1.3 Entscheidungsvariablen und Zielfunktion

Betrachtet man die formulierten Annahmen und Einschränkungen, so ist die einzige Veränderliche die Abfahrtszeit einer Linie in ihrem Startpunkt; sind die Abfahrtszeiten aller Linien bekannt, ist der gesamte Fahrplan eindeutig bestimmt. Die Entscheidungsvariable ist also ein Vektor aus den Abfahrtszeiten aller Linien. Die Zielfunktion bildet diesen Vektor auf die daraus resultierende Wartezeit ab.

3.2 Das mathematische Modell

Die hier zitierten Autoren formulieren die gestellte Aufgabe als *quadratic semi-assignment problem* QSAP. Ziel ist hier die Minimierung der Gesamtübergangszeit als der Summe aller individuellen Übergangszeiten innerhalb einer gegebenen Betriebszeit.

Hierzu nimmt man die Menge aller betrachteten Linien $L = \{1, \ldots, m\}$ und die Menge aller möglichen Abfahrtszeiten, hier repräsentiert durch die Menge der Indizes einer Aufzählung der Abfahrtszeiten $D = \{1, \ldots, n\}$. Die „Kosten-“, d.h. die Wartezeitenmatrix c_{ihjk}, die die Gesamtübergangszeit am Knoten der Linien i und j unter der Voraussetzung der Abfahrtszeit h der Linie i und der Abfahrtszeit k der Linie j angibt, wird als symmetrisch angenommen; sie ergibt sich als Summe der individuellen Wartezeiten aller Passagiere in diesem Knoten. Nun führt man noch die binäre Variable

$$x_{ih} = \begin{cases} 1 & \text{wenn (Abfahrtszeit) } h \in D \text{ der Linie } i \in L \text{ zugeordnet ist} \\ 0 & \text{in allen anderen Fällen} \end{cases}$$

ein, so daß das QSAP formuliert wird als:

$$\text{Minimiere } Z(x) = \sum_{i=1}^{m} \sum_{h=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{n} c_{ihjk} x_{ih} x_{jk} \tag{3.1}$$

unter den Einschränkungen

$$\sum_{h=1}^{n} x_{ih} = 1 \qquad i = 1, \ldots, m \tag{3.2}$$

$$x_{ih} \in \{0, 1\} \qquad i = 1, \ldots, m; h = 1, \ldots, n \tag{3.3}$$

Für jede Linie i existiert eine Menge $DT(i) = 1, \ldots, t_i$ der möglichen Abfahrtszeiten, wobei t_i die Dauer des Abfahrens der gesamten Linie (*basic cycle time*, Liniendauer) angibt. Beim Modellfall gleicher Anzahl möglicher Startzeiten $t_i = n$ für alle Linien ergibt sich als Umfang des Problems der Term „Anzahl der Linien $\times$ Liniendauer“; korrekter ist die Aufsummierung, wenn statt n $n(i) = |DT(i)|$ verwendet wird.

Auf andere mögliche Formulierungen wie die Modifizierung des QSAP-Modells durch die gewichtete Einführung von die Unsicherheit im wirklichen Leben widerspiegelnder Faktoren oder die Verwendung vom QSAP gänzlich unabhängiger Modelle wird von den Autoren nicht weiter eingegangen.

3.3 Die Lösung

3.3.1 TS-Grundlagen

Ausgangspunkt ist eine (zufällig ausgewählte oder durch andere Verfahren, z.B. *regret heuristic*, voroptimierte) Anfangslösung. Ein Zug bestehe dann aus der Veränderung von genau zwei Attributen (*paired-attribute moves*). So beschreibt der Zug $(\overline{x_{ih}}, x_{ik})$ den Wechsel der Startzeit der Linie i von h nach k. Man beachte, daß $\overline{x_{ih}}$ nicht das Negieren der binären Variable x_{ih} darstellt, sondern den Wechsel von Eins auf Null; entsprechend bezeichnet x_{ih} hier nicht die Identitätsoperation, sondern den Wechsel von Null auf Eins.

Aus einer Liste möglicher Züge wird dann derjenige ausgewählt, der die größte Verbesserung resp. die geringste Verschlechterung der Zielfunktion erreicht. Zur Vermeidung einer Endlosschleife kann hier bereits ein einfacher TS-Mechanismus implementiert werden: Ausgeführte Züge als Repräsentation der bereits durchlaufenen Lösungen erscheinen in einer laufenden Liste, daraus wird eine attribut-basierte Tabu-Liste gewonnen, in der die Komplementärattribute der laufenden Liste aufgenommen werden, da ja diese zu bereits geprüften Lösungen führen würden. So entsteht aus der Liste der möglichen Züge eine um die tabuisierten Züge verminderte Liste der zulässigen Züge. Eine allgemeine Formulierung des Algorithmus könnte so aussehen:

Gegeben sei eine mögliche Lösung x^* mit einem Zielfunktionswert z^*, setze $x := x^*$ mit $z(x) = z^*$.

Iteration:

solange das Abbruchskriterium nicht erfüllt ist, *wiederhole*:

1. Wähle den besten zulässigen Zug, der x nach x' überführt und füge seine Attribute zur laufenden Liste hinzu.
2. Aktualisiere die Tabu-Liste.
3. Tausche aus: $x := x', z(x) = z(x')$, *wenn* $z(x) < z^*$ *dann* $z^* := z(x), x^* := x$ *endewenn*

endesolange

Ergebnis: x^* ist die beste aller untersuchten Lösungen mit dem Zielfunktionswert $z(x^*)$.

3.3.2 Tabulisten und Suchstrategien

Entscheidend für die Bewertung des obigen Algorithmus ist das Führen der Tabu-Liste. Grundsätzlich wird der Tabu-Liste eine *recency based* Liste als Kurzzeitgedächtnis zugrundeliegen. Diese wird dann durch verschiedene *frequency based* Strukturen ergänzt, durch die auch langfristigere Effekte Eingang finden. Durch diese Kombination wird eine Balance zwischen Intensivierung und Diversifikation (GLOVER UND LAGUNA, 1993, S. 109 ff) erreicht.

Intensivierung besteht in der Beibehaltung von in der Vergangenheit „erfolgreichen“ Attributen. Formeller dargestellt, handelt es sich um die Suche in Regionen um bisher gefundene Optima herum, mit dem Risiko, viel Zeit an ein nicht globales Optimum zu verwenden. Diversifikation hingegen verfolgt das Ziel, möglichst Lösungen mit gänzlich anderen Attributen zu untersuchen; hierdurch wird ein größerer Ausschnitt des Lösungsraums durchsucht, allerdings u.U. weitmaschiger. Das Risiko besteht darin, daß zu früh von einem nur knapp verfehlten globalen Optimum wieder weggesprungen wird. Beide Strategien haben also ihre Vor- und Nachteile, so daß eine gute TS-Implementation eine Abwägung zwischen beiden vornehmen muß.

Implementiert wird eine Gewichtung zwischen diesen beiden Strategien GLOVER UND LAGUNA (1993, S.111) folgend durch die Einführung einer *Penalty/Incentive Func-*

tion PI, die entweder auf *from*- oder *to*-Attribute angewandt wird. Hierfür wird eine Funktion auf zwei entsprechende Attribute eines beabsichtigten Zuges angewandt, die ein Maß für die bisherige Häufigkeit des Attributs liefert. Die PI-Funktion liefert dann einen aus beiden Häufigkeiten gewonnenen (Minimum, Maximum, Durchschnitt,...), evtl. um einen Schwellenwert verminderten Wert, der, falls positiv, beibehalten, ansonsten auf Null gesetzt wird. PI-Funktionen über *to*-Attributen begünstigen eine Intensivierungsstrategie, solche über *from*-Attributen eine Diversifikationsstrategie.

3.3.3 Dynamisches Tabulistenmanagement

In den bisher betrachteten Konzepten wurde die Gültigkeitsdauer von Tabus als konstant angenommen. Solche Konzepte nennt man statisch; sie werden auch als *Tabu Navigation Method* TNM bezeichnet. Verbesserungen der Effektivität von TS sind möglich durch den Übergang zum dynamischen Tabulistenmanagement.

DADUNA UND VOSS behandeln zwei Konzepte dynamischen Tabulistenmanagements: die *Cancellation Sequence Method* CSM und die *Reverse Elimination Method* REM.

CSM betrachtet die laufende Liste als eine Kandidatenliste von Attributen, deren Komplemente tabu-aktiv werden dürfen. Wenn ein Attribut sein Komplement auf der Kandidatenliste findet, wird dieses aus der Liste getilgt. Alle Attribute zwischen dem gelöschten und dem gerade hinzugefügten bilden eine *cancellation sequence*; die hier enthaltenen Attribute dürfen bei künftigen Zügen nicht getilgt werden. Züge, die zu einer Umkehrung eines Attributs der *cancellation sequence* führen würden, sind somit tabu. Die Effektivität von CSM hängt von der Wahl der Gültigkeitsdauer eines Tabus ab. CSM stellt ein hinreichendes, aber nicht notwendiges Kriterium dar.

REM beruht auf der Idee, daß eine bereits geprüfte Lösung nur dann noch einmal besucht werden kann, wenn sie in der Nachbarschaft der aktuellen liegt. Daher wird die Tabuliste nach jedem Zug neu gebildet. Hierfür wird die laufende Liste rückwärts nach Zügen durchsucht, die tabu gesetzt werden müssen. Dabei wird eine *residual cancellation sequence* RCS, die leer initialisiert wird, nur mit den Attributen gefüllt, deren Komplemente nicht in der Sequenz enthalten sind. Ist ein Komplement enthalten, wird es gelöscht. Wenn nach Prüfen aller bisheriger Züge die RCS Attribute enthält, die durch einen Zug umgekehrt werden können, ist dieser Zug tabu. Die Gültigkeitsdauer des Tabus beträgt eine Iteration.

REM wurde durch VOSS zu REM2 verfeinert (VOSS, 1993). Hier sind zusätzlich alle gemeinsamen Nachbarn der gerade betrachteten und einer bereits geprüften Lösung verboten, da diese Nachbarn aufgrund der Wahl eines besten nicht-tabuisierten Nachbarn in der Vergangenheit bereits implizit getestet wurden. REM wie REM2 sind hinreichende und notwendige Kriterien.

3.4 Ergebnisse

3.4.1 Modellrechnungen

DADUNA UND VOSS (1993) haben in Modellrechnungen TS und *simulated annealing* sowie die verschiedenen Strategien von TS miteinander verglichen. Hierfür wurden kleine Modellnetzwerke von zehn bis vierundzwanzig Linien mit einer identischen Fahrzeit von fünf bzw. zehn Minuten erstellt und mittels *regret heuristic* bestimmte Startlösungen in den *simulated annealing*- bzw. TS-Algorithmus eingegeben. Verglichen wurde die durchschnittliche, minimale und maximale Abweichung der ermittelten Lösung von der bestmöglichen bei verschiedenen Iterationstiefen sowie die Anzahl der Lösungsläufe, die gegenüber der Startlösung eine Verbesserung erreichten. Alle TS-Methoden zeigen hier eine erheblich verbesserte Effizienz gegenüber *simulated annealing*. So wird nur bei TS die bestmögliche Lösung in der erlaubten Rechenzeit gefunden; die Anzahl der Lösungsläufe mit Verbesserung lag bei allen TS-Methoden mehr als doppelt so hoch.

3.4.2 Fallstudien

Weiterhin wurden die Ergebnisse von drei Fallstudien dargestellt, die in verschiedenen Städten durchgeführt wurden. Die erforderlichen Daten wurden durch Fahrgastzählungen zu repräsentativen Zeiten erhoben.

Im ersten Fall wurden 28 Linien mit 15 Übergangsknoten und pro Linie unterschiedlichen Fahrtdauern, die auch von der Tageszeit abhingen, unter weiteren Beschränkungen betrachtet. Die Daten wurden zweimal ausgewertet: Einmal ohne Gewichtung der Wartezeiten, so daß diese wie gemessen in die Berechnungen eingingen, und einmal mit Gewichtung, wobei alle Wartezeiten von null bis drei Minuten als Null gewertet wurden. Da bereits die durch *regret heuristic* ermittelte Startlösung nahe am Optimum lag, konnte sie durch alle TS-Methoden nur noch leicht verbessert werden, wobei bei gewichteten Wartezeiten die durchschnittliche Wartezeit sogar leicht anstieg. Die Authoren stellten keine Unterschiede zwischen TNM, CSM und REM heraus.

In der zweiten Fallstudie wurden 54 Linien mit 38 Übergangsknoten und nur zwei unterschiedlichen Linienfahrtdauern unter weiteren Einschränkungen betrachtet. Da hier der Nachtdienst optimiert wurde, entfielen tageszeitabhängige Fahrtdauern. Die Ergebnisse gleichen denen der ersten Fallstudie; hier schnitt von allen TS-Methoden TNM am besten ab.

In einer dritten Fallstudie ergaben sich noch stärkere Beschränkungen durch die sternförmige Anordnung der Linien auf einen zentralen Bahnhof hin. Diese zusätzlichen Beschränkungen erlaubten nur noch geringe Verbesserungen.

Die TS-Algorithmen zur Fahrplansynchronisation wurden auch in ein Projekt eingebettet, in dem neue Buslinien in ein existierendes ÖPV-Netz einer Großstadt eingefügt wurden. Das Ziel war in diesem Projekt praxisgerechter nicht nur die Minimierung kumulierter Wartezeiten, sondern auch anderer Faktoren wie z.B. Kosten. In dieser komplexen Situation traten auch Beschränkungen auf, die nicht im Modell erfaßt werden konnten. Daher wurden bei der Synchronisation mehrere gute Lösungen, die von der besten gefundenen Lösung maximal einen vorgegebenen Prozentsatz abwichen, gesucht, die dann von Hand auf die Erfüllung anderer Kriterien wie Kostensenkung überprüft wurden; waren hier starke Ersparnisse möglich, so wurde auch eine knapp suboptimale Lösung bei der Synchronisierung akzeptiert.

Literaturverzeichnis

[Banerjea-Brodeur u. a. 1998] BANERJEA-BRODEUR, M. ; CORDEAU, J.-F. ; LAPORTE, G. ; LASRY, A.: Scheduling linen deliveries in a large hospital. In: *J. Oper. Res. Soc. (UK)* 49 (1998), Nr. 8, S. 777–80

[Brandao und Mercer 1998] BRANDAO, J.C.S. ; MERCER, A.: The multi-trip vehicle routing problem. In: *J. Oper. Res. Soc.* 49 (1998), Nr. 8, S. 799–805

[Clarke u. a. 2000] CLARKE, John ; HARMAN, Mark ; JONES, Bryan ; LUMKIN, Mary ; REES, Kearton ; ROPER, Marc ; SHEPPERD, Martin: *The application of Metaheuristic Search Techniques to Problems in Software Engineering*. 2000. – URL `http://www.brunel.ac.uk/~csstmmh2/seminal/landscapes/home.html`

[Daduna u. a. 1993] DADUNA, Joachim R. (Hrsg.) ; BRANCO, Isabel (Hrsg.) ; PAIXAO, José M. P. (Hrsg.): *Computer-Aided Transit Scheduling*. Bd. *430*. Springer, 1993. (Lecture notes in economics and mathematical systems; Proceedings of the 6th international workshop on Computer-Aides Cheduling of Public Transport, Lisbon, Portugal)

[Daduna und Voß 1993] DADUNA, Joachim R. ; VOSS, Stefan: Practical Experiences in Schedule Synchronization. In: DADUNA, Joachim R. (Hrsg.) ; BRANCO, Isabel (Hrsg.) ; PINTO, José M. (Hrsg.): *Computer-Aided Transit Scheduling* Bd. 430, Springer, 1993, S. 39–55

[Gambardella 2000] GAMBARDELLA, Luca M.: *Metaheuristics Network*. 2000. – URL `http://www.idsia.ch/luca/abstracts/papers/corso_vrp_metanet.pdf`

[Glover 1992] GLOVER, F.: Multilevel tabu search and embedded search neighbourhoods for the travelling salesman problem. In: *ORSA J. on Computing* (1992)

[Glover und Laguna 1993] GLOVER, Fred ; LAGUNA, Manuel: Tabu Search. In: REEVES, Colin R. (Hrsg.): *Modern heuristic techniques for combinatorial problems*. Blackwell Scientific Publications, Oxford, 1993, Kap. 3, S. 70–150

[Google 2001] GOOGLE: *Google deutsch*. 2001. – URL `http://www.google.de/search?hl=de&safe=off&q=%22Tabu+search%22+application+-sex&btnG=Google-Suche&meta=`

[Higgins 1998] HIGGINS, A.: Scheduling of railway track maintenance activities and crews. In: *J. Oper. Res. Soc.* 49 (1998), Nr. 8, S. 1026–33

[Laguna u. a. 1993] LAGUNA, M. ; BARNES, J.W. ; GLOVER, F.: Intelligent Scheduling with Tabu Search: An Application to Jobs with Linear Delay Penalties an Sequence Dependent Setup Costs and Times. In: *Journal of Applied Intelligence* 3 (1993), S. 159–172

[Langenscheidt-Verlag 2001] LANGENSCHEIDT-VERLAG: *Langenscheidts Fremdwörterbuch*. 2001. – URL `http://www.langenscheidt.de/deutsch/index.html`

[Lee u. a. 2000] LEE, Youngho ; KIM, Youhwan ; KIM, Jeongheon: *A broadband access network problem in designing mass market multi-media networks*. 2000. – URL `http://informs.scu.edu/papers/4237.pdf`

[Lim u. a. 2000] LIM, Yun F. ; LAU, Hoong C. ; LIU, Qizhang: Diversification of Search Neighbourhood via Constraint-based local Search and its application to Vehicle Routing. In: *Operations Research in the Millenium*, 2000

[Malek u. a. 1989] MALEK, M. ; GURUSWAMY, M. ; PANDYA, M. ; OWENS, H.: Serial and parallel simulated annealing and tabu search algorithms for the travelling salesman problem. In: *Annals of Ops. Res.* 21 (1989), S. 59–84

[Sexton u. a. 1999] SEXTON, R. ; ALLIDAE, B. ; DORSEY, R. ; JOHNSON, J.: *Global Optimization for Artificial Neural Networks: A Tabu Search Application*. 1999. – URL `citeseer.nj.nec.com/251353.html`

[Srivastava 1998] SRIVASTAVA, B.: An effective heuristic for minimising makespan on unrelated parallel machines. In: *J. Oper. Res. Soc.* 49 (1998), Nr. 8, S. 886–94

[Voß 1993] VOSS, Stefan: Tabu Search: Applications and prospects. In: DU, D.-Z. (Hrsg.) ; PARDALOS, P. (Hrsg.): *Network Optimization Problems*. World Scientific, Singapore, New Jersey, London, Hong Kong, 1993, S. 333–353

[Widmer und Hertz 1989] WIDMER, M. ; HERTZ, A.: A new heuristic method for the flow shop sequencing problem. In: *EJOR* 41 (1989), S. 186–193