

# Effizientes Lösen von Schedulingproblemen mit "Vorschlägen und Vertauschen"

Rainer Herrler

University of Würzburg, Department for Artificial Intelligence,  
herrler@informatik.uni-wuerzburg.de  
<http://ki.informatik.uni-wuerzburg.de/>

**Abstract** Schedulingprobleme sind von großer praktischer Bedeutung und sie können als ein Spezialfall von Zuordnungsproblemen betrachtet werden. Deshalb eignet sich die "Vorschlägen und Vertauschen"-Methode [10], eine heuristische Problemlösungsmethode für Zuordnungsprobleme. Die betrachtete Methode ist sehr flexibel und konnte durch den Einsatz problemspezifischen Wissens bereits in der Vergangenheit einige komplexe Schedulingprobleme in akzeptabler Zeit lösen. In vorliegender Arbeit wurden einige Versuche zu einer weiteren Effizienzsteigerung der Vorschlägen und Vertauschen-Methode vorgenommen.

## Zuordnungsprobleme

Ausgangsmengen eines Zuordnungsproblems sind eine Menge von *Nachfragern* und eine Menge von *Anbietern*. Gesucht ist eine Abbildung der Menge der Nachfrager auf die Menge der Anbieter, wobei verschieden gewichtete *Randbedingungen* berücksichtigt werden sollen. Ich nenne diese Randbedingungen im folgenden auch *Restriktionen*, da sie die Menge der gültigen Abbildung einschränken. Jede Abbildung der Menge der Nachfrager auf die Menge der Anbieter heißt *Zuordnung*. Jeder Nachfrager muß mit einem Anbieter versorgt werden. Anbieter müssen nicht notwendigerweise einem Nachfrager zugeteilt werden, können aber auch zugleich mehreren Nachfragern zugeteilt werden. In diesem Falle spricht man von einer *1-n-Zuordnung*. Der Spezialfall, daß jeder Anbieter genau einem Nachfrager zugeteilt wird, heißt *1-1-Zuordnung*.

Praktische Beispiele für Zuordnungsprobleme sind die Schulstundenplanung, die Ressourcenbelegungsplanung oder die Erstellung von Dienstplänen für das Pflegepersonal in Krankenhäusern (kurz Schwesternplanung).

Bei der Ressourcenbelegungsplanung sind die Nachfrager die Fertigungsschritte, die sich aus den Aufträgen ergeben. Die Anbieter für diese Fertigungsschritte sind unbelegte Zeitintervalle der vorhandener Maschinen. Randbedingungen bestehen z.B. durch Liefertermine oder aufgrund einer bestimmten Reihenfolge, in der die Fertigungsschritte durchgeführt werden müssen.

Bei der Dienstplanerstellung müssen Dienstsichten auf das zur Verfügung stehende Personal verteilt werden. Die zu leistenden Dienstsichten sind also die Nachfrager des Zuordnungsproblems, die Arbeitstage des Personals sind die Anbieter. Typische Randbedingungen sind beispielsweise gesetzliche Bestimmungen zur maximalen Anzahl von Nachtschichten in Folge oder Urlaubswünsche der Angestellten.

Wie die Beispiele schon andeuten, ist es sinnvoll, zwischen *harten Restriktionen* und *weichen Restriktionen* zu unterscheiden. Harte Restriktionen müssen unbedingt eingehalten werden, um eine gültige Lösung zu erhalten. Weiche Restriktionen sind zusätzliche Optimierungskriterien. Um auch die Wichtigkeit einer Restriktion berücksichtigen zu können, sind diese gewichtet.

Bei den aufgeführten Beispielen für Zuordnungsprobleme handelt es sich um Schedulingprobleme. Zuordnungsprobleme umfassen den Problembereich Scheduling, sie sind jedoch nicht darauf beschränkt. Scheduling ist die zeitliche Zuweisung von Aktivitäten zu Ressourcen [12]. Bei einem Raumzuweisungsproblem, bei dem die Mitarbeiter einer Arbeitsgruppe auf bestimmte Büros verteilt werden müssen [9], fehlt die zeitliche Komponente, es können jedoch bei Betrachtung als Zuordnungsproblem prinzipiell dieselben Problemlösemethoden verwendet werden.

## Vorschlagen und Vertauschen

### Motivation

Auch in sehr einfachen Problemdomänen ist die Anzahl der kombinatorisch möglichen Zuordnungen schon sehr groß. Bei  $n$  Nachfragern und  $a$  Anbietern gibt es  $a^n$  Zuordnungsmöglichkeiten (ungeachtet ihrer Zulässigkeit durch die Restriktionen). Sofern es nur lineare Randbedingungen gibt, wie statische Präferenzen zwischen Anbietern und Nachfragern, so gibt es exakte Lösungsverfahren, die in polynomieller Zeit eine optimale Lösung finden. Die Situation ändert sich sehr schnell, wenn die Problemstellung komplexe Randbedingungen enthält. (z.B. dynamische Präferenzen, die von zuvor erfolgten Zuweisungen, abhängen, bzw. Randbedingungen, die Gruppen von Zuweisungen betreffen). Eine Beispiel für eine solche komplexe Randbedingung bei der Schwesternplanung ist die Vorschrift, daß nicht mehr als 4 Nachtschichten in Folge auftreten dürfen. Durch solche komplexen Randbedingungen wird das Problem NP-hart, ist also in der Praxis durch einen exakten Algorithmus nur in exponentieller Zeit lösbar.

Für einen praktikablen Einsatz verwendet man deshalb heuristische und approximative Verfahren. Im Zentrum dieser Arbeit steht eine solche Problemlösungsmethode, die

Vorschlagen und Vertauschen-Methode [11]. Im speziellen werden die Auswirkungen verschiedener Realisierungsvarianten auf die Effizienz untersucht.

### Der Algorithmus

Der Vorschlagen und Vertauschen-Algorithmus (kurz V&V) ist in vielen Punkten der menschlichen Vorgehensweise bei der Lösung von Zuordnungsproblemen nachempfunden. Es erzeugt eine initiale Lösung durch schrittweise Zuteilung von Nachfragern zu Anbietern und optimiert diese im Anschluß. Um dabei entstandene Restriktionsverletzungen möglichst frühzeitig zu beseitigen, werden schon bei der Konstruktion der initialen Lösung in jedem Schritt lokale Optimierungen durchgeführt. Wenn eine Verletzung nicht in einem bestimmten Zeitlimit beseitigt werden kann, wird die beste in dieser begrenzten Zeit gefundene Konstellation gewählt und mit der Lösungskonstruktion fortgefahren. Diese Verletzungen können bei der anschließenden globalen Optimierung beseitigt werden, oder bleiben als Restverletzung bei der gefundenen Lösung.

Im folgenden werden die einzelnen Schritte des Algorithmus anhand von Abbildung 1 detaillierter beschrieben.

Die Problemlösung beginnt mit der Auswahl eines noch nicht zugeordneten Nachfragerobjektes (Schritt 1 der Abbildung). Diese Auswahl kann entweder zufällig geschehen, oder nach einer, vom Benutzer vorgegebenen Strategie. Hier kann also bereits problemspezifisches Wissen einfließen, das den Berechnungsaufwand zum Finden einer Lösung reduziert.

Im zweiten Schritt wird ein geeignetes, freies Angebotsobjekt für den Nachfrager ausgesucht und zugeordnet. Das passende Angebotsobjekt, d.h. das, das durch seine Zuteilung möglichst wenige Randbedingungen verletzt, wird aus den *Vorschlägen*, einem weiteren problemspezifischen Wissensselement ausgesucht.

Für die durchgeführte Zuteilung erfolgt in Schritt 3 eine Überprüfung aller zugehörigen Randbedingungen. Wurden Randbedingungen verletzt, so versucht der Algorithmus diese Verletzungen in Schritt 4 aufzulösen. Falls keine lokale, d.h. die aktuelle Zuteilung betreffende Verletzung, besteht, fährt der Algorithmus mit der Konstruktion der initialen Lösung fort.

Schritt 4, auch genannt die *lokale Optimierung* versucht, die Verletzungen der zuletzt durchgeführten Zuteilungen durch Vertauschungen mit anderen Nachfragern zu beseitigen. Sie Suche nach einer Verbesserung wird durch ein vorgegebenes Zeitkontingent beschränkt.

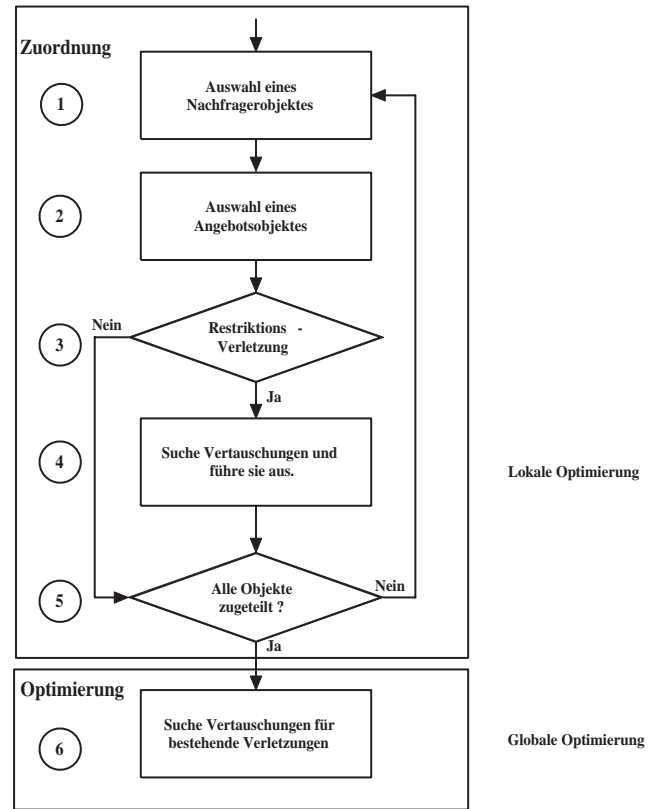
Wenn alle Nachfrager zugeteilt sind (Test durch Schritt 5), existiert eine vollständige Zuordnung, die jedoch noch Restriktionsverletzungen enthalten kann. Das sind zum einen Verletzungen, die die lokale Optimierung nicht beseitigen konnte, als auch Verletzungen von Restriktionen, deren Berücksichtigung während der Konstruktion noch nicht sinnvoll ist. Dies ist der Fall bei Randbedingungen, die durch Zuteilung weiterer bisher nicht zugeteilter Nachfrager erfüllt werden können. Da diese an die Lösungskonstruktion anschließende Optimierung Verletzungen zu allen Zuteilungen betrifft, wird sie *globale Optimierung* genannt.

## Der Shellbaukasten COKE

COKE [8] ist eine Expertensystemshell für Zuordnungsprobleme, die auf der Vorschlägen und Vertauschen-Methode basiert. Um eine Anwendung für eine Problemdomäne zu erstellen, muß der Wissensingenieur im wesentlichen folgende Aufgaben lösen. Er muß

- die beteiligten Objekte (Nachfrager und Anbieter) modellieren,
- die Restriktionen formulieren,
- das strategische Problemlösewissen in Auswahl-, Vorschlags- und Vertauschungsregeln abbilden.

- und ein angepaßtes Benutzerinterface für das Problem entwerfen.



**Figure1.** Illustration aus [1] zum Vorschlägen und Vertauschen-Algorithmus.

Die hier vorgeschlagenen Varianten wurden in COKE implementiert und bildeten hier die Grundlage für die angestrebten Effizienzuntersuchungen. Auf COKE basierend wurden bereits mehrere Anwendungen realisiert, darunter Problemlöser für das Raumbelungsproblem, die Schulstundenplanung und die Ressourcenbelegungsplanung. In neuerer Zeit kamen ein PEPSI [3], ein Prototyp zur Einsatzplanung des Zugbegleitpersonals der Deutschen Bahn und NURSE [2], eine Anwendung zur Dienstplanerstellung von Pflegepersonal in Krankenhäusern hinzu. Diese beiden Anwendungen wurden zur Evaluation verwendet.

## Effizienzuntersuchungen

### Effizienzkriterien

Jeder der Algorithmusschritte zeichnet sich nicht nur durch mehrere Realisierungsmöglichkeiten sondern auch durch verschiedene Möglichkeiten zur Einbringung problemspezifischen Wissens aus. Diese Varianten beeinflussen maßgeblich die Effizienz.

Effizienz beinhaltet auf Zuordnungsprobleme gesehen eine möglichst hohe Qualität der Lösungen. Die Anforderungen an Lösungen sind in den verschiedenen Anwendungsbereichen sehr unterschiedlich, gemeinsam ist ihnen jedoch die Forderung nach Einhalten der harten Randbedingungen und die Optimierung der weichen Randbedingungen. Effizienz muß in Relation zu den aufgewandten Mitteln gesehen werden. Ein Algorithmus, der hinsichtlich der Lösung sehr leistungsfähig ist, d.h. die beste Lösung findet, kann bei realen Problemen unpraktikabel sein, da die nötigen Mittel nicht aufgebracht werden können. Bei der Zuordnung spielen die Mittel Speicher, Zeit und Wissen eine wesentliche Rolle.

Speicher ist der begrenzende Faktor vieler Lösungsverfahren. Zudem ist er in realen Umgebungen immer endlich und kann nicht beliebig ausgebaut werden. Zeit kann dagegen zwar theoretisch beliebig hinzugenommen werden, ist in der Praxis jedoch auch ein bedeutender Leistungsfaktor. Wichtig ist, daß man das Ergebnis abwarten kann, bzw. daß das Ergebnis nicht schon veraltet ist. So ist die Zeit zwischen Bekanntwerden der Anforderungen und dem Termin, an dem eine Lösung benötigt wird, üblicherweise beschränkt. Ein gutes Beispiel ist die Ressourcenbelegungsplanung, wo sich häufig die Anforderungen an einen Plan ändern und deshalb schnell eine neue Lösung benötigt wird. Eine weitere Ressource ist das Wissen. Zusätzliches Wissen über das Problem kann die Leistungsfähigkeit eines Systems zur Lösung verbessern. Der Aufwand, der in Wissenserwerb und Berechnung steckt, muß mit der gewonnenen Lösungsqualität im Verhältnis gesehen werden.

## Frühere Effizienzuntersuchungen

Frühere Effizienzuntersuchungen [7] des V&V-Verfahrens testeten Varianten der Algorithmusschritte 1 und 2. Sie unterschieden sich im Aufwand, der für die Auswahl der Nachfrager und getrieben wird.

Analog zu den Heuristiken bei Constraint-Satisfaction-Problemen hat es sich als vorteilhaft herausgestellt, bei der Auswahl des Nachfragerobjektes zunächst Nachfrager mit geringem Freiheitsgrad (die *most constrained variable*) zuzuteilen und bei der Auswahl des Anbieterobjektes auf die Erhaltung möglichst vieler Freiheitsgrade zu achten (den *least constraining value* zu wählen).

Je nach Problemdomäne kann es unterschiedlich wichtig und effektiv sein, eine aufwendige Auswahlstrategie zu verwenden. Die Ergebnisse aus [7] belegen, daß es für einfache Zuordnungsprobleme weitgehend unerheblich ist, welche der Varianten gewählt wird. Je komplexer die Zuordnungsprobleme jedoch werden, desto besser ist es, viel Rechenzeit und Wissen in diese frühen Algorithmusschritte zu investieren. Dagegen ist es in diesem Fall sehr schwierig oder gar unmöglich, eine schlechte Lösung durch Vertauschungen in späten Zuordnungsschritten zu reparieren. Dies wurde an einem komplexeren Fall der Schulstundenplanung gezeigt.

### Effizienzuntersuchungen der Optimierung

Noch sehr unzureichend wurden bisher verschiedene Möglichkeiten, die sich bei der Realisierung der lokalen Optimierung bieten, untersucht. Zum einen können hier verschiedene Basis-Suchalgorithmen zur Suche nach Verbesserungen eingesetzt werden, zum anderen kann auch die Art der Vertauschungen konkretisiert werden. Die Vertauschungen können beispielsweise zufällig ausgesucht werden, es bietet sich jedoch an, hier problemspezifisches Wissen zu verwenden. Im folgenden wird etwas genauer aufgeschlüsselt, aus

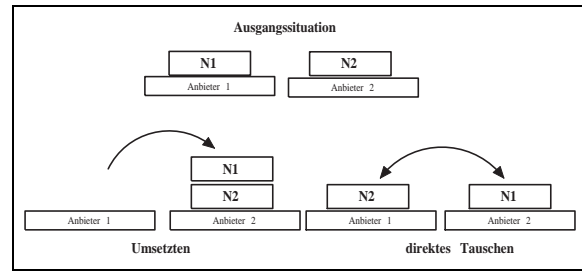
welchen Teilen die lokale Optimierung besteht und wie man sie realisieren kann.

Grundlage der Optimierung ist eine Suche im Suchraum der partiellen Zuordnungen<sup>1</sup>, bei der genau dieselben Nachfrager zugeteilt sind, die zu Beginn der Suche zugeteilt waren. Nicht festgelegte Bestandteile der Suche sind noch

- die Nachbarschaften,
- die Bewertungsfunktion und
- die Art des verwendeten Suchalgorithmus.

**Nachbarschaften** Die Nachbarschaften im Suchraum werden durch die Vertauschungen angegeben. Eine Berücksichtigung aller möglichen Vertauschungen bei Zuordnungsproblemen mit größeren Nachfragermengen hätte verheerende Auswirkungen auf den Verzweigungsgrad des Suchbaumes, also muß man sich hier problemspezifisch auf eine Auswahl beschränken.

Jede Veränderung einer partiellen Zuordnung kann als Kombination mehrerer *elementarer Veränderungen* betrachtet werden. Diese sind das *Zurückziehen* einer Zuteilung und das *Setzen* eines zuvor zurückgezogenen Nachfragers. Zwei Möglichkeiten, die Nachbarschaften zu definieren, sind das *Umsetzen* und das *direkte Tauschen*, die beide in Abbildung 2 am Beispiel veranschaulicht werden. Läßt man auch die elementare Veränderungen als Nachbarn zu, so erweitert man damit den Suchraum. Die Elemente des Suchraums sind dann auch partielle Zuordnungen der Lösungsraumes. In diesem Fall betrachtet man genaugenommen partielle Zuordnungen von partiellen Zuordnungen. Im folgenden gehen wir für das bessere Verständnis vom einfachsten Fall aus, daß jede Vertauschung ein Umsetzen eines Nachfragers auf einen neuen Anbieter ist.



**Figure 2.** Mögliche Veränderungen an einer Zuordnung. Die Veränderungen definieren benachbarte Zustände im Suchraum.

Es wurde bereits festgestellt, daß die Verfolgung aller möglichen Vertauschungen bei der Suche zu kostenintensiv ist. Nun soll konkretisiert werden, wie dieser Aufwand reduziert werden kann. Wie bei der intuitiven manuellen Problemlösung, sollen die Nachfrager umgesetzt werden, die die aktuelle Zuteilung "stören", das kann z.B. bei der Dienstplanung eine zeitlich benachbarte Schicht sein, oder bei der Ressourcenbelegungsplanung ein Auftrag, der zum gegebenen Zeitpunkt um die gewünschte Ressource konkurriert. Diese *Störenfriede* sind problemspezifisch unterschiedlich. Die Wissenskomponente der Problemlösemethode enthält deshalb Störenfriedregeln, die nach einer verletzten Zuteilung potentiell störende Nachfrager angeben. Folgende beide Möglichkeiten unterscheiden sich im Aufwand des Wissenserwerbs:

- *Zuteilungsabhängige Störenfriede* Die Störenfriede werden abhängig vom Problem und der Zuteilung angegeben. In diesem Falle ist für eine Problemdomäne nur eine Regel zur Bestimmung der Störenfriede nötig. Diese Funktion bestimmt für eine Zuordnung potentielle Störenfriede, z.B. benachbarte Zuteilungen.
- *Verletzungsabhängige Störenfriede* Die Störenfriede werden abhängig vom Problem, der Zuteilung und der verletzten

<sup>1</sup> Bei einer *partiellen Zuordnung* sind (noch) nicht alle Nachfrager zugeteilt, die Element der Ausgangsnachfragermenge sind.

Randbedingung angegeben.

Für jede Randbedingung existiert eine eigene Regel zur Bestimmung der Störenfriede. Der erhöhte Aufwand im Wissenserwerb macht auf der anderen Seite eine speziellere Angabe der Störenfriede möglich. In der Dienstplanung kann eine Verletzung der Randbedingung, daß nicht mehr als drei Nachtschichten in Folge auftreten dürfen, durch gezieltes Vertauschen der ersten oder letzten Schicht aufgelöst werden.

Beide Varianten und ein generischer, nicht wissenbasierter Ansatz, bei dem alle zugeteilten Nachfrager als Störenfriede gesehen werden, wurden in COKE realisiert. In beiden untersuchten Anwendungen wurde die Qualität der Lösung durch die verletzungsabhängige Angabe der Störenfriede bei einer gleichzeitigen Verkürzung der Laufzeit verbessert (Vergleiche Tabellen 1 und 2). Es lohnt also, mehr Wissen, bzw. spezielleres Wissen über das Problem zu einzusetzen. Die generische Variante kann ebenfalls das Ergebnis verbessern, der erhöhte Zeitbedarf steht jedoch mit dem Gewinn nicht im Verhältnis

Es ist noch zu erwähnen, daß durch den wissenbasierten Ansatz der Suchraum nicht mehr so stark vernetzt ist. Es gibt keine Garantie mehr, daß ein Pfad im Suchraum vom gegenwärtigen Zustand zum Optimum besteht. Im allgemeinen wird die Suche jedoch gerichteter und deshalb effizienter. Von der Qualität der Störenfriedenfunktionen hängt es ab, ob die Suche weiterhin vollständig ist und wie schnell eine Lösung gefunden werden kann.

Das Wissen um die störenden Nachfrager kann auf verschiedene Weise eingesetzt werden. Es wurde bereits angedeutet, daß ein Vertauschen aus zwei elementaren Schritten, dem Zurückziehen und dem Setzen eines Störenfriedes, besteht. Nach dem Zurückziehen entsteht ein Zwischenzustand mit zurückgezogenen Nachfragern. Diese werden bei der weiteren Suche gesetzt. Dabei können wiederum

Verletzungen entstehen, zu denen Störenfriede ermittelt werden können. Dieses "Schneeballsystem" kommt zur Ruhe, wenn entweder keine Verletzung mehr besteht oder die Zeitschranke der lokalen Optimierung überschritten wird. Zum Zurückziehen der Störenfriede werden hier drei Varianten vorgestellt:

– *Störenfriede werden einzeln zurückgezogen*

Es wird jeweils nur ein Störenfried zurückgezogen. Bei seinem Setzen werden erneut die ihn störenden Nachfrager für den nächsten Suchschritt ermittelt. Nachteil hierbei ist, daß Gruppen von Nachfragern, die eine Verletzung verursachen, nicht aufgelöst werden können.

– *Störenfriede werden alle gemeinsam zurückgezogen*

Nach einer Verletzung werden alle Störenfriede gleichzeitig zurückgezogen. Vor dem nächsten Suchschritt müssen erst wieder alle zurückgezogenen Nachfrager zugeteilt werden. Von einem Suchschritt auf den nächsten wird hier kein einzelnes Umsetzen vorgenommen, sondern eine komplexe Vertauschung, an der mehrere Nachfrager beteiligt sind.

– *Störenfriede werden in Gruppen zurückgezogen*

Die zuvor genannte Strategie, alle Nachfrager zurückzuziehen und in allen möglichen Kombinationen neu zu setzen, erzeugt sehr viele Nachfolgezustände, was die Suche wiederum ineffektiv macht. Häufig jedoch reicht auch das einzelne Zurückziehen völlig aus. Ein Kompromiß ist, diese Entscheidung wieder in die Wissenskomponente zu verlagern und dort störende Gruppen von Nachfragern anzugeben (siehe Beispiel in Abbildung 3).

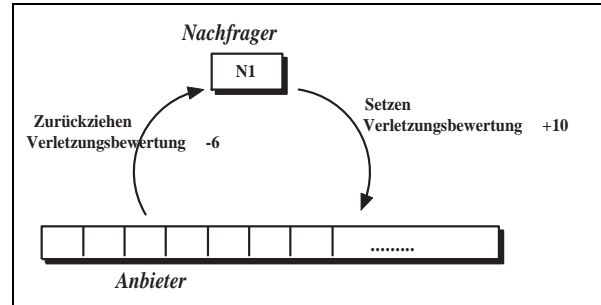
einfache Störenfriedliste:  
 $M_1 = \{N_1, N_2, N_3, N_4, N_5, \dots, N_i\}$   
 Störenfriede in Gruppen:  
 $M_2 = \{\{N_1\}, \{N_2\}, \{N_3\}, \{N_4\}, \{N_5\}, \dots, \{N_i\}\}$   
 $M_3 = \{\{N_1, N_2, N_5\}, \{N_1\}, \{N_2\}\}$

**Figure3.**  $M_1$  ist ein Beispiel für eine einfache Störenfriedmenge. Es können entweder alle Nachfrager einzeln oder alle gemeinsam zurückgezogen werden.  $M_2, M_3$  sind Beispiele für eine erweiterte Wissensrepräsentation der Störenfriede. Bei der Korrektur mit  $M_3$  werden zunächst  $N_1, N_2$  und  $N_5$  zurückgezogen, um den Konflikt zu beheben. Erst danach werden die Nachfrager wieder zugeteilt.

Im Rahmen der Effizienzuntersuchungen in COKE wurden die Varianten *einzelnes Zurückziehen* und *alle Zurückziehen* implementiert und miteinander verglichen. Der dritte Vorschlag bleibt noch an einem geeigneten Problem zu untersuchen. Bei den untersuchten Varianten verbesserte das gemeinsame Zurückziehen das Optimierungsergebnis bei PEPSI mit angemessenem Zeitzuwachs, bei NURSE jedoch explodierte die Rechenzeit und es konnte keine Verbesserung erzielt werden (Vergleich Tabellen 1 und 2). PEPSI und NURSE unterscheiden sich dadurch, daß die Menge der betrachteten Nachfrager bei der Schwesternplanung erheblich höher ist. Diese Option verspricht also bei einfacheren Zuordnungsproblemen mit kleinen Nachfragermengen eine Leistungssteigerung, ist jedoch bei Problemen mit größeren Nachfragermengen nicht anwendbar.

**Bewertungsfunktion** Für informierte Suchstrategien benötigt man eine Bewertungsfunktion, die es ermöglicht die Suche heuristisch zu steuern. Mit der Verletzungsbewertung der Restriktionen, ist uns eine solche heuristische Funktion vorgegeben. Ihr Minimum ist zu-

gleich auch das Ziel der Suche. Im allgemeinen ist eine exakte, vollständige Bewertung der partiellen Zuordnung in jedem Schritt zu kostspielig. Eine Lösung ist, statt der vollständigen Bewertung die relative Veränderungsbewertung als Heuristik für die Suche zu verwenden. Abbildung 4 illustriert diese Idee.



**Figure4.** Beim Zurückziehen eines Nachfragers von einem Anbieter verändert sich die Verletzungsbewertung der Restriktionen, die diese Zuteilung betreffen. Bei einer Neuzuteilung tritt erneut eine Verletzung auf. Zur Beurteilung interessiert nur die Veränderung der Gesamtbewertung durch diesen Umsetzungsvorgang.

Leider sind diese Veränderungsbewertungen nicht vollständig konsistent zu der exakten und vollständigen Bewertung aller Restriktionen. Die Methode hat den Nachteil, daß Verletzungsveränderungen in "benachbarten"<sup>2</sup> Zuteilungen nicht berücksichtigt werden. Dies muß im Hinblick auf die Rechenzeit hingenommen werden. Es gibt zwei naheliegende Varianten der Veränderungsbewertung, die im folgenden anhand des Beispiels in Abbildung 5 beschrieben werden:

– *Veränderung am Störenfried*

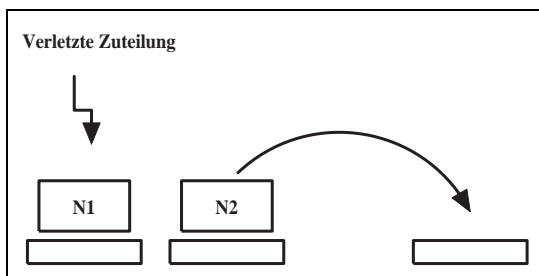
Die Veränderungsbewertung setzt sich zusammen aus einer Bewertungsverringerung in Höhe der ursprünglichen Verletzung des Störenfrieds ( $N_2$  im Beispiel) und einer Bewertungserhöhung um die

<sup>2</sup> In vielen Problemdomänen kann man Distanzen auf der Anbietermenge definieren. Restriktionsverletzungen gelten häufig für eine Gruppe von Zuteilungen auf benachbarten Anbietern.

erzeugten Verletzungen durch die erneute Zuteilung. Ob die ursprüngliche Verletzung, deren Korrektur das Ziel war, beseitigt oder verringert wurde, spiegelt sich in dieser Veränderungsbewertung nicht wieder.

- *Veränderung am zu korrigierenden Nachfrager*

Alternativ kann man beim Zurückziehen des Nachfragers  $N2$  als Bewertungsverbesserung auch die Differenz zwischen der neuen Bewertung von  $N1$  und der alten Bewertung von  $N2$  verwenden. Im Fokus der Optimierung bleibt die eigentlich zu korrigierende Verletzung, da ihr Ausmaß in die Bewertung einfließt. So wird zielgerichteter eine Korrektur der Ausgangsverletzung angestrebt.



**Figure 5.** Das Bild beschreibt folgende Situation: Die Zuteilung des Nachfragers  $N1$  ist verletzt.  $N2$  wird als Störenfried angegeben und wird deshalb umgesetzt. Das Zurückziehen von  $N2$  hebt die Verletzungen an  $N2$  auf, beeinflusst jedoch auch die Verletzungsbewertung an  $N1$ .

In den Versuchen mit dem Dienstplanungsanwendungen für die Deutsche Bahn und die Krankenhäuser hat sich die zweite Variante als die deutlich bessere gezeigt (Vergleiche Tabelle 1 und 2).

**Suchalgorithmus** Als Basissuchalgorithmus für die Optimierung hat sich in COKE die Bestensuche bewährt. Die Anforderungen an

den Suchalgorithmus sind, daß er die vorhandene heuristische Funktion benutzt und lokale Minima überwinden kann. Interessante Alternativen wären deshalb im wesentlichen die Beam-Suche [6] oder Simulated Annealing.

Der Vorteil der Beam-Suche gegenüber Bestensuche, liegt darin, daß sie speichereffizienter ist. Der Engpaß bei der Lösung der bisher betrachteten Probleme in COKE war jedoch die Ressource Zeit. Untersuchungen über die Effektivität der Beamsuche bei der Optimierung von Zuordnungsproblemen - eventuell in komplexeren Problem domänen - stehen also noch aus.

Experimente mit Simulated Annealing in COKE [5] zeigten, daß Simulated Annealing auch ein geeigneter Ansatz zur Optimierung von Zuordnungsproblemen sein kann, die Bestensuche hat sich in der langfristigen Anwendung aber dennoch durchgesetzt. Versuche mit reinem Hillclimbing und Tiefensuche wiesen eindeutig schlechtere Optimierungsergebnisse auf [4], was aber im Hinblick auf die oben gestellten Anforderungen (informiertes Suchverfahren, Überwindung lokaler Minima) nicht überraschend ist.

## Zusammenfassung

"Vorschlagen und Vertauschen" hat sich in der Anwendung für viele Zuordnungs- und Schedulingprobleme als sehr geeignet erwiesen. Die Stärken liegen in der leichten Anpassbarkeit an eine Problem domäne, in den Möglichkeiten zur Einbringung von Expertenwissen und der interaktiven Planung. Auf der anderen Seite ist es ein sehr allgemeines Paradigma und bedarf einer genaueren Spezifikation der Einzelschritte, bevor Aussagen über die Leistungsfähigkeit getroffen werden können. Es wurden verschiedene Varianten für die lokale Optimierung, bzw. die Suche nach Vertauschungen vorgestellt und verglichen. Dabei gab es sowohl Varianten, die sich ganz allgemein als gut erwiesen, als auch Varianten, die bei speziellen Eigenschaften von Problem domänen z.B. kleinen



Nachfragemengen besonders sinnvoll sind. Aufgrund der Untersuchungen der vorliegenden Arbeit ist es gelungen den Problemlöser in COKE effizienter zu machen. Das beinhaltet sowohl einen Performancegewinn als auch bessere Optimierungsergebnisse, die nun erzielt werden können. Davon profitieren die bereits existierenden, auf COKE basierenden Anwendungen. In einigen Fällen konnte die Verletzungsbewertung sogar auf weniger als ein Drittel der ursprünglichen Bewertung reduziert werden. Bei der Zugbegleiterplanung, bei der früher gelegentlich interaktiv eingegriffen werden mußte, können jetzt problemlos ohne Benutzereingriff optimale Lösungen gefunden werden. Die Erfolge bei der Schwesternplanung waren nicht ganz so groß; zwar konnten Verbesserungen bei der Optimierungsleistung erzielt werden, die Ergebnisse blieben dennoch suboptimal. Dies liegt jedoch an dem frühen Entwicklungsstadium, in dem sich die Wissenskomponente des Nurse-Schedulers befindet. Die Hoffnung, den geringem Aufwand im Wissenserwerb allein mit algorithmischen Verbesserungen ausgleichen zu können, hat sich nicht erfüllt. Weitere Arbeit an der Wissensbasis kann hier die Planung verbessern.

## References

1. Ciske Busch: *Spezifikation eines kooperativen Werkzeugs zur Lösung komplexer Zuordnungsprobleme*. Diplomarbeit, Universität Würzburg, 1997.
2. Frank Forster: *Automatische Generierung von Oberflächen zur interaktiven Lösung von Zuordnungsproblemen*. Diplomarbeit, Universität Würzburg, 1999.
3. Rainer Herrler: *PEPSI ein Tool zur interaktiven Dienstplanung des BORD-Personals der Deutschen Bahn AG*. Studienarbeit, Universität Würzburg, 1999.
4. Rainer Herrler: *Effizienzuntersuchungen in der Vorschlagen und Vertauschen-Methode für Zuordnungsprobleme*. Diplomarbeit, Universität Würzburg, 2000. Diplomarbeit.
5. Franziska Klügl: *Optimierung von Zuordnungsproblemen durch Simuliertes Ausglühen und Genetische Algorithmen*. In: INFIX Richter et al.

(Hrsg.): *XPS '95*, Seiten 119–136, St. Augustin, 1995.

6. Peter Norvig: *Artificial Intelligence Programming*. Morgan Kaufmann Publishers, 1992.
7. Karsten Poeck and Frank Puppe: *COKE: Efficient Solving of Complex Assignment Problems with the Propose-and-exchange Method*. In *Proc. IEEE Conference on Tools with Artificial Intelligence*, pages 136–143, 1992.
8. Karsten Poeck: *Konzeption und Implementierung eines problemspezifischen Expertensystemwerkzeugs zur Zuordnung mit der Vorschlagen-und-Vertauschen-Strategie*. Diplomarbeit, Universität Karlsruhe, April 1990.
9. Karsten Poeck: *Solving the YQT-Problem with the Assignment Shell COKE*. Arbeitspapiere der GMD 630, GMD, April 1991.
10. Frank Puppe: *Problemlösungsmethoden in Expertensystemen*. Springer, 1990.
11. Frank Puppe: *Problemlösungsmethoden in Expertensystemen*. Springer, Heidelberg, Berlin, 1990.
12. Jürgen Sauer: *Knowledge-Based Systems Techniques and Applications in Scheduling*. to appear in: Leondes, T.L. (Ed.): *Knowledge-Based Systems Techniques and Applications*, Academic Press, San Diego.

PEPSI DP 714	Bewertung	Laufzeit hh:mm:ss	#Zustände
Standard, Zuteilungsabhängige Störenfriede			
Variante 1	1721,60	07:15	23940
Verletzungsabhängige Störenfriede			
Variante 2	452,00*	06:26	18339
alle Nachfrager sind Störenfriede			
Variante 3	541,60*	22:20	62509
Standard, Störenfriede einzeln zurückziehen			
Variante 1	1721,60	07:15	23940
Störenfriede gemeinsam zurückziehen			
Variante 4	1331,20	04:02	24796
Standard, Veränderung am zu korrigierenden Nachfrager			
Variante 1	1721,60	07:15	23940
Veränderung am Störenden			
Variante 5	3660,00	07:25	41489

**Tabelle 1:** Gemessene Werte bei der Erstellung des PEPSI-Plans 714. Dieser Dienstplan enthält 55 Nachfrager und ist einer der schwierigsten Dienstpläne der zur Verfügung stehenden Testdaten. Bei einigen Testkonfigurationen dieser Versuchsreihe gelingt es dennoch, ohne manuellen Eingriff einen brauchbaren Dienstplan zu erstellen.

NURSE 1.1.2000	Bewertung	Laufzeit hh:mm:ss	#Zustände
Standard, Zuteilungsabhängige Störenfriede			
Variante 1	4708,80	00:34	5491
Verletzungsabhängige Störenfriede			
Variante 2	1467,80	44:59	743104
alle Nachfrager sind Störenfriede			
Variante 3	4708,80	01:12:34	1362032
Standard, Störenfriede einzeln zurückziehen			
Variante 1	4708,80	00:34	5491
Störenfriede gemeinsam zurückziehen			
Variante 4	4708,80	00:44	10635
Standard Veränderung am zu korrigierenden Nachfrager			
Variante 1	4708,80	00:34	5491
Veränderung am Störenden			
Variante 5	10382,04	00:30	5147

**Tabelle 2:** Verschiedene Konfigurationen bei der lokalen Optimierung eines Nurse-Scheduling-Planes. Der Dienstplan geht über einen Zeitraum von drei Wochen und enthält ca. 200 Dienstsichtenen.