

ConPlan/SIEDAplan: Personaleinsatzplanung als Constraintproblem

Harald Meyer auf'm Hofe, Enno Tolzmann

Das ConPlan-Projekt, das am Deutschen Forschungszentrum für Künstliche Intelligenz in Zusammenarbeit mit der SIEDA Systemhaus für intelligente EDV-Anwendungen GmbH Kaiserslautern durchgeführt wurde, beschäftigte sich mit der Repräsentation der Personaleinsatzplanung im Pflegedienst von Krankenhäusern als *hierarchical constraint satisfaction problem (HCSP)*. Ergebnis des Projekts ist u.a. das System SIEDAplan zur Dienstplangenerierung in einem Krankenhausinformationssystem, das gegenwärtig im DRK Krankenhaus Neuwied getestet wird.

1 Einleitung

Das ConPlan-Projekt¹ am Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) hatte zum Ziel, die Verwendbarkeit weicher Constraints anhand einer anspruchsvollen Anwendung zu erforschen. Als Maßstab diente die Dienstplangenerierung für den Pflegedienst von Krankenhäusern mit den Anforderungen, wie sie von der SIEDA GmbH in Zusammenarbeit mit dem DRK Krankenhaus in Neuwied und der Uniklinik Mainz erhoben wurden. Personaleinsatzplanung als eine Instanz des *time tabling*-Problems betrifft die Zuordnung von Ressourcen, in diesem Falle die Arbeitszeit von Pflegekräften, zu bestimmten Aufgaben über eine bestimmte Planungsdauer. Die Dienstplangenerierung für den Pflegedienst zeichnet sich dabei durch zwei besondere Eigenschaften aus:

(1) Die konkreten Problemstellungen variieren sehr stark durch viele Parameter. So müssen unterschiedliche Arbeitsverträge der Beschäftigten berücksichtigt werden. Wegen des Mangels an hochqualifiziertem Pflegepersonal möchten die Krankenhäuser auch Teilzeitarbeitskräfte effektiv einsetzen. Zeitkonten sind zu berücksichtigen. Überstunden sind so weit wie möglich zu vermeiden, da sie Kosten verursachen. Das Krankenhaus möchte in der Lage sein, flexibel auf unterschiedliches Arbeitsaufkommen reagieren zu können. Eine vorab erstellte Urlaubsplanung muß berücksichtigt werden. Durch die Vielzahl an Freiheitsgraden ist ein festes Vorgehensmodell für die Dienstplanung schwer vorstellbar.

(2) An einen Dienstplan im Pflegedienst werden unterschiedlichste Anforderungen gestellt, die sich im allgemeinen nur zum Teil erfüllen lassen. Diese betreffen den Ausgleich von Überstunden, präferierte Schichtrhythmen, arbeits- und tarifrechtliche Bestimmungen wie etwa den Mutter- und Jugendschutz, sowie Präferenzen des Pflegepersonals selbst.

Dienstpläne werden gegenwärtig in wochenlanger Vorbereitung von Hand erstellt. Das System SIEDAplan hingegen erstellt Dienstpläne von mindestens vergleichbarer Qualität auf einfachen Pentium-Rechnern in wenigen Minuten.

Harald Meyer auf'm Hofe studierte bis 1994 Informatik an der Universität Kaiserslautern und ist seitdem als wissenschaftlicher Mitarbeiter am DFKI tätig. Seine Forschungsinteressen gelten der Constraint-Verarbeitung, *fuzzy*-Methoden und wissensbasierten Konfigurations- und Design-Systemen.

Enno Tolzmann studierte von 1987 bis 1993 Informatik an der Universität Kaiserslautern. Er ist Gründungsmitglied der SIEDA Systemhaus für intelligente EDV-Anwendungen GmbH (Mai 1993) und seit November 1994 ihr Geschäftsführer.

¹Das ConPlan-Projekt wurde durch die Stiftung Rheinland-Pfalz für Innovation unter dem Förderkennzeichen 836-38 62 61/98 unterstützt.

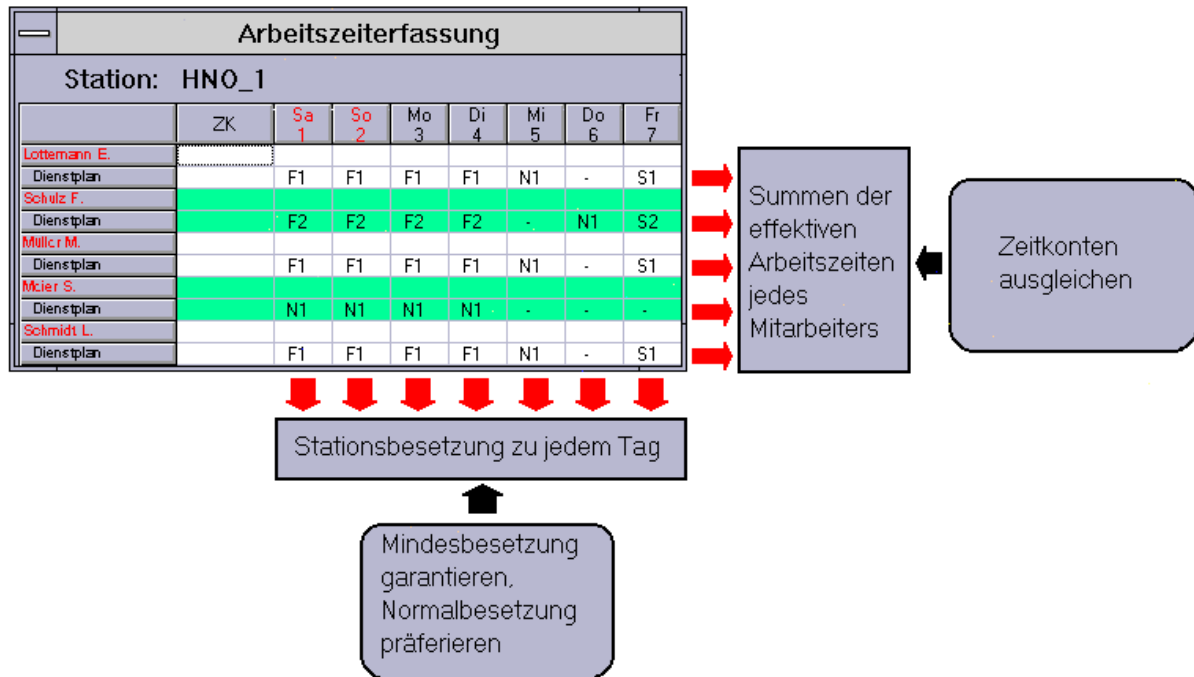


Abbildung 1: Abhängigkeiten bei der Dienstplangenerierung.

2 Modellierung

Um ein Problem der Personaleinsatzplanung als ein CSP zu formulieren, müssen zunächst die Variablen, ihre Domänen und abschließend die Randbedingungen, die an den Dienstplan gestellt werden, spezifiziert werden. Aus zwei Gründen bietet es sich für die Dienstpläne des Pflegedienstes in Krankenhäusern an, für jede Person und jeden Tag des Planungszeitraumes (1 Monat) eine Variable vorzusehen, der dann die möglichen Schichten zugewiesen werden. Zum einen sind die möglichen Schichten, die eine Pflegekraft absolvieren soll, durch innerbetriebliche Absprachen vorgegeben. Zum anderen kodiert diese Form der Zuordnung auf elegante Weise die Randbedingung, daß jede Pflegekraft aus Abrechnungsgründen an jedem Tage eine (Frei-) Schicht absolvieren muß. Es ergeben sich Probleme mit 600 bis 800 Variablen. Zugewiesen werden unterschiedliche Früh-, Spät- und Nachtschichten, die sich durch Anfangs-, End- und Pausenzeiten und die effektiv geleistete Arbeitszeit unterscheiden. In Dienstplänen werden sie z.B. durch **F1** (Frühschicht), **S1** (Spätschicht) und **N1** (Nachtschicht) dargestellt. Darüber hinaus werden verschiedene Formen von Freischichten, z.B. für Urlaub (**UL**) und Ausgleichstage (—), wegen unterschiedlicher Abrechnungsmodi bzgl. Arbeitskonten und Arbeitszeit unterschieden. Die grundsätzlichen Abhängigkeiten zwischen diesen Schichten in einem Dienstplan werden in Abbildung 1 schematisch dargestellt. Die Zeilen

im Dienstplan betreffen den Schichtrhythmus ein und derselben Person. Randbedingungen betreffen die Verträglichkeit von aufeinanderfolgenden Schichten, die Verträglichkeit mit sogenannten Arbeitszeitmodellen (über längere Zeiträume präferierte Schicht rhythm en) und den Ausgleich von Zeitkonten. Randbedingungen über die Spalten des Dienstplanes betreffen die Besetzung der Station, die z.B. von der Belegung der Station abhängt. Jede dieser Randbedingungen liegt sowohl in einer obligatorischen, «harten» als auch in einer schwerer zu erfüllenden aber optionalen, «weichen» Form vor. So muß beispielsweise die Mindestbesetzung der Station unbedingt gewährleistet sein, während die (größere) Normalbesetzung präferiert wird.

Die Constraints zur Stationsbesetzung und der effektiven Arbeitszeit (Führung des Zeitkontos) haben ca. 20 bzw. ca. 30 lokale Variablen und damit unhandlich große Extensionen. Zu ihrer effektiven Propagierung müssen spezielle Eigenschaften der Relationen ausgenutzt werden, um Domänen unbelegter Variablen zu propagieren.

Das wesentliche Problem der Modellierung war jedoch die Repräsentation und Auswertung weicher Constraints. Die Constraints der Personaleinsatzplanung wurden in eine Constrainthierarchie eingeordnet [1,3] mit den folgenden Hierarchieebenen: Die Gewährleistung der Minimalbesetzung bei Einhaltung der Zeitkontengrenzen ist oberstes Ziel. Es folgt das Erreichen der Normalbesetzung und der Ausgleich der Zeitkonten. Da nicht ausgeglichene Zeitkonten

Dienstplangenerierung																			
Station: Innere-01		von: 01.11.1996 bis: 30.11.1996																	
	ZK	Fr 1	Sa 2	So 3	Mo 4	Di 5	Mi 6	Do 7	Fr 8	Sa 9	So 10	Mo 11	Di 12	Mi 13	Do 14	Fr 15	Sa 16	So 17	M 18
Hübner Günther	-15h24	N1	N1	N1	N1	N1	N1	N1	N1	-	-	*	-	-	-	N1	N1	N1	N
Leibig Markus	-1h24	F1	-	F1	F1	S1	S1	S1	S1	S1	S1	S1	*	F1	F1	F1	*	-	S
Löffler Rita	-8h54	S1	-	-	F1	F1	F1	F1	F1	F1	UL	UL	UL	UL	UL	UL	-	-	F
Mischnick Eva	3h06	N1	-	-	-	-	-	-	N1	N1	N1	N1	N1	N1	N1	N1	*	-	*
Müller Heidrun	-2h24	S1	S1	S1	S1	*	F1	F1	F1	-	-	S1	S1	S1	S1	S1	S1	S1	S
Schmidt Bettina	-1h24	F1	F1	S1	S1	S1	*	S1	S1	-	-	F1	F1	F1	F1	F1	F1	S1	S

	ZK	Fr 1	Sa 2	So 3	Mo 4	Di 5	Mi 6	Do 7	Fr 8	Sa 9	So 10	Mo 11	Di 12	Mi 13	Do 14	Fr 15	Sa 16	So 17	M 18
Hübner Günther	-15h24	-	N1	N1	N1	N1	N1	N1	*	-	F1	F1	*	-	-	-	N1	N1	N
Leibig Markus	-1h24	F1	-	F1	F1	F1	S1	S1	S1	S1	S1	*	F1	F1	F1	F1	*	-	F
Löffler Rita	-8h54	N1	-	F1	F1	F1	F1	F1	F1	F1	UL	UL	UL	UL	UL	UL	*	-	F
Mischnick Eva	3h06	-	-	-	-	-	-	-	N1	N1	N1	N1	N1	N1	N1	N1	*	-	*
Müller Heidrun	-2h24	S1	S1	S1	S1	*	F1	F1	F1	-	-	S1	S1	S1	S1	S1	S1	S1	S
Schmidt Bettina	-1h24	F1	F1	S1	S1	S1	*	S1	S1	-	-	F1	F1	F1	F1	F1	F1	F1	S

Abbildung 2: Initialer und durch lokale Suche verbesserter Dienstplan (Ausschnitte).

Kosten verursachen, stehen sie möglichst weit oben in der Hierarchie. Direkt in der nächsten Hierarchiestufe wird die Verträglichkeit mit den Arbeitszeitmodellen berücksichtigt. Die niedrigste Hierarchiestufe stellen Einsatzwünsche der Pflegekräfte dar. Innerhalb dieser Hierarchieebenen sind die Constraints nach zusätzlichen Gewichten angeordnet. Bei der Modellierung des Problems kann somit auf einfache Art spezifiziert werden, daß z.B. möglichst viele Einsatzwünsche von Pflegekräften berücksichtigt werden – jeder Wunsch hat ein Gewicht, das Gesamtgewicht der Wünsche wird optimiert – wobei der Ausgleich von Zeitkonten jedoch vorranglich behandelt wird – die entsprechenden Constraints liegen in einer höheren Hierarchiestufe.

3 Das Suchverfahren

Probleme mit 600 bis 800 Variablen können im allgemeinen nicht vollständig durchsucht werden. Als Alternative bieten sich lokale Suchverfahren der folgenden Art an:

- lokale_Suche(V, C, D)*
1. berechne initiale Belegung aller Variablen in V;
 2. wähle eine Variablenmenge $V' \subset V$, deren Belegung verbessert werden soll;
 3. wenn $V' = \{ \}$ dann liefere aktuelle Belegung als Ergebnis;
 4. ansonsten ermittle neue Belegung der Variablen in V' ;
 5. gehe zu Anweisung 2;

Die Schleife der Anweisungen 2 bis 5 kann jederzeit unterbrochen werden, um ein mehr oder weniger gutes Ergebnis zu liefern. Lokale Suchverfahren unterscheiden sich zum einen darin, ob in Anweisung 4 mit einer gewissen Wahrscheinlichkeit eine Verschlechterung der aktuellen Belegung erlaubt wird – wie im GSAT-Algorithmus [6] – oder eine strikte *hill climbing*- Strategie verwendet wird – wie im *minimize conflicts* Algorithmus [5]. Von ersteren

ist bekannt, daß sie bei geeigneter Wahl dieser Wahrscheinlichkeit gegen eine optimale Lösung des Problems konvergieren. Von letzteren ist bekannt, daß sie schneller zu einer Verbesserung der Lösung führen [7]. Diese Eigenschaft ist für die dargestellte Anwendung entscheidend. Das Dienstplangenerierungssystem muß schnell eine gute Antwort liefern, um den Benutzern z.B. durch Einfügen neuer Anforderungen Eingriffsmöglichkeiten in die Planung zu bieten. Hingegen bleiben *hill climbing*-Algorithmen in lokalen Optima hängen. Ein wesentlicher Faktor für die Güte der errechneten Lösung ist die Güte der initialen Belegung der Variablen aus Anweisung 1. Kann hier bereits effizient eine relativ gute Belegung berechnet werden, so hat es der lokale Suchalgorithmus natürlich leichter. Die speziellen Möglichkeiten der Constraintverarbeitung [3,4] – insbesondere von Constrainthierarchien – sollten genutzt werden, um so schnell wie möglich einen guten Dienstplan zu berechnen. Dies geschieht an zwei Stellen: Der initialen Belegung der Variablen (im Schema Anweisung 1) und der lokalen Verbesserung (Anweisung 4).

Prospektive Constraintverfahren (Propagierung von Domänen nicht belegter Variablen) werden verwendet, um eine möglichst gute initiale Belegung der Variablen zu erhalten. Dabei werden diejenigen Schichten zugewiesen, die die wenigsten Constraintverletzungen erwarten lassen. Ergebnis ist ein Dienstplan wie der zuoberst in Abbildung 2 dargestellte. Die helle Schattierung zeigt an, daß die geplante Stationsbesetzung nicht genau der gewünschten entspricht. Die dunkle Schattierung zeigt nicht akzeptable Stationsbesetzungen an. An zwei Sonntagen wurden die Mindestbesetzungen nicht eingehalten, weil keine Frühschichten eingeplant wurden.

Ein lokales Suchverfahren soll nun diese initiale Belegung verbessern. Um in Anweisung 4 des

Schemas der lokalen Suche für eine Teilmenge V' der Variablen eine optimale Belegung zu ermitteln, soll eine Variante des *branch&bound*-Algorithmus zur vollständigen Suche verwendet werden.

Ein Blick auf den initialen Dienstplan in Abbildung 2 zeigt, daß sich die Kandidaten für die zu verbessernde Variablenbelegungen V' durch einige Heuristiken stark einschränken lassen. Für einige Probleme wie z.B. die falsche Stationsbesetzung am 17.11. genügt die Änderung einer einzigen Schicht. Um jedoch eine Unterbesetzung wie am 10.11. des Monats zu beheben, muß in der Regel an einem anderen Tage eine Überbesetzung gefunden werden wie z.B. am 8.11. Hier besteht V' sinnvollerweise aus Variablen zu diesen beiden Tagen. Diese letztere Heuristik erinnert an das Suchverfahren zur ressourcen-orientierten Konfigurierung [2]. Um eine Unterbesetzung an einem bestimmten Tage zu vermeiden, muß an einem anderen Tage eine Überbesetzung gefunden werden, die den erforderlichen Mehrbedarf an Arbeitsstunden kompensieren kann. Solche und ähnliche Heuristiken veranlassen die lokale Suche, dort nach Optimierungen einer Variablenbelegung zu suchen, wo sie auch zu finden sind. Dabei werden zunächst diejenigen heuristisch bestimmten Kandidaten für die Menge V' ausprobiert, die die wenigsten Variablen enthalten. Daher werden schnell durchführbare Möglichkeiten zur Verbesserung eines Dienstplanes zuerst ergriffen.

Der untere Dienstplan in Abbildung 2 zeigt das Ergebnis dieser Optimierung für das Beispiel. Die *nicht akzeptablen* Stationsbesetzungen konnten hinreichend verbessert werden. Auch die meisten der *akzeptablen* Mängel des initialen Dienstplanes konnten vermieden werden.

Dieser Dienstplan ist ein Beispiel dafür, daß mit den gegebenen Ressourcen häufig nicht allen Forderungen nachgekommen werden kann. An einigen Tagen lassen sich Überbesetzungen nicht vermeiden, weil mehr Personal als benötigt verfügbar ist und die Zeitkontenuntergrenzen bereits erreicht sind. Da auch in diesen Fällen ein Dienstplan berechnet werden muß, ist die Modellierung der Dienstplangenerierung als Optimierungsproblem unumgänglich.

4 Fazit

Das vorgestellte Problem der Dienstplangenerierung im Pflegedienst von Krankenhäusern wurde durch eine problemspezifische Form der lokalen Suche gelöst. Bei der üblichen Herangehensweise an derartige Probleme durch Algorithmen der Baumsuche wie dem *branch&bound* ist die aufwendige Untersuchung

vollständiger Suchbäume auf Symmetrien und die explizite Programmierung von *choice points* notwendig, um den Suchraum handhabbar zu machen. Die vorgestellten Heuristiken zur lokalen Suche sind wesentlich leichter zu finden und auch stabiler gegenüber Änderungen der Modellierung durch Einfügung neuer Forderungen an den Dienstplan.

Ziel weitergehender Forschung ist es, geeignete Anpassungen lokaler Suchverfahren aus der Problemstellung als Constraintproblem automatisch zu ermitteln. Diese Erweiterung ist für viele Problemklassen wie z.B. die Konfigurierung notwendig.

Literatur

1. Alan Borning, Bjorn Freeman-Benson und Molly Wilson, *Constraint Hierarchies*, Lisp and Symbolic Computation, 5, 223-270, Kluwer Academic Publishers, 1992.
2. Michael Heinrich. *Ressourcenorientiertes Konfigurieren*. KI - Künstliche Intelligenz. Seiten 11-15. Nummer 1. 1993
3. Harald Meyer auf'm Hofe und Bidjan Tschaitshian. *PCSPs with hierarchical constraint orderings in real-world scheduling applications*. Aus: Michael Jampel, Eugene Freuder und Michael Maher (Hrsg.) *Workshop notes on the CP'95 workshop on over-constrained systems*.
4. Harald Meyer auf'm Hofe. *Partial satisfaction of constraint hierarchies in reactive and interactive configuration*. Aus: Walter Hower und Zsófia Ruttkay (Hrsg.) *Non-standard Constraint Processing*, Workshop Notes of the ECAI'96 Workshop W27, 1996.
5. Steven Minton, Mark Johnston, Andrew Philips und Philip Laird. *Minimizing conflicts: a heuristic repair method for constraint satisfaction problem and scheduling problems*. Artificial Intelligence, 58:161-205, 1992.
6. Bart Selman, Hector Levesque und David Mitchell. *A new method for solving hard satisfiability problems*. Aus: AAAI-92: Proceedings of the 10th National Conference on AI, Seiten 440-446, 1992.
7. Rick Wallace und Eugene Freuder. *Heuristic methods for over-constrained constraint satisfaction problems*. Aus: Michael Jampel, Eugene Freuder und Michael Maher (Hrsg.) *Workshop Notes CP95 Workshop on Over-Constrained Systems*, Cassis, France, 1995.