

Verwendung von constraint-basierten Modellen zur Lösung von Bauablaufproblemen mit Hilfe der Constraint-Solver-Bibliothek firstCS.

Use of Constraint-based Modelling to Solve Construction Scheduling Problems Applying the Constraint Solver Library firstCS

Ramez Edris, Volkhard Franz,
Universität Kassel, Kassel (Germany)
ramez.edris@first.fraunhofer.de

Armin Wolf
Fraunhofer FIRST, Berlin (Germany)

Abstract: Nowadays, the Constraint Programming is a significant and steadily growing field of research and application. In practice, the Constraint Programming paradigm is successfully applied to solve combinatorial problems, in particular in scheduling, simulation and diagnosis, and also in many industrial areas (HOFSTEDT, WOLF 2007, S.VII). In this paper it is shown how the constraints of the constraint solver library firstCS (WOLF 2006) can be used efficiently for modeling of construction sequence problems. Moreover, the suitable constraint models as well as the searching heuristics of firstCS are used and discussed with the help of an application case.

1 Einleitung

Bei der Erstellung von Bauablaufplänen in Bauprojekten im Allgemeinen und im Hochbau im Besonderen ist eine Vielzahl von projektspezifischen Randbedingungen zu berücksichtigen. Bei der Erstellung von Ablaufplänen untergliedern sich die Aufgaben der Planer in drei Teilaufgaben: Bestimmung der Vorgänge, Festlegung der Beziehungen zwischen den Vorgängen und Bestimmung der Dauer der Vorgänge (HUHNT, ENGE 2007, S. 33). Zur Bestimmung der Terminpläne wird die Netzplantechnik als klassische Herangehensweise zur Ermittlung von Terminplänen benutzt (BRANDENBERGER, RUOSCH 1993, S. 43).

Bei der Netzplantechnik können nur die gegenseitigen Abhängigkeitsbeziehungen zwischen einzelnen Vorgängen dargestellt werden. In der Praxis sind aber auch andere Arten von Abhängigkeiten sehr wichtig, z.B. Abhängigkeiten, die

wünschenswert, aber nicht unbedingt einzuhalten sind, oder Abhängigkeiten, die sich auf mehrere Vorgänge gleichzeitig beziehen. In dieser Arbeit wird eine Modellierungsmethode vorgeschlagen, die alle diese Abhängigkeiten berücksichtigen kann und dabei die wünschenswerten Abhängigkeiten von den zwingend notwendigen Abhängigkeiten unterscheidet.

2 Abhängigkeitsbeziehungen zwischen den Bauaufgaben in Bauabläufen (Constraints)

Im Allgemeinen lassen sich die Abhängigkeitsbeziehungen in technologische und kapazitative Abhängigkeiten unterteilen. Die kapazitiven Abhängigkeiten ergeben sich aus den Einschränkungen der Einsatzmittel sowie den zur Verfügung stehenden Ressourcen. Die technologischen Abhängigkeiten hängen von den ausgewählten Bauverfahren, der Baukonstruktion und anderen Randbedingungen ab. Die technologischen Abhängigkeiten können weiter in zwingende und zweckmäßige Abhängigkeiten unterteilt werden (NAGEL u.a. 1990). In diesem Beitrag werden die technologischen Abhängigkeiten analysiert und nach (ROSSI u.a. 2006) als lokale, die wieder in Soft- und Hard-Constraints unterteilt werden, sowie auch globale Constraints klassifiziert. Weiterhin werden die Gewerke als einzelne Ressourcen betrachtet, so dass jedes Gewerk nur eine Arbeitsgruppe hat.

2.1 Lokale Constraints der technologischen Abhängigkeiten zwischen den einzelnen Vorgängen

Die lokalen Constraints beziehen sich auf eine bestimmte Anzahl von Bauaufgaben. Sie können weiter in zwingende Abhängigkeiten (Hard-Constraints) und zweckmäßige Abhängigkeiten (Soft-Constraints) unterteilt werden:

- Hard-Constraints beschreiben die zwingenden zeitlichen Abhängigkeiten und ordnen die zugehörigen Bauaufgaben in eine zwingende Reihenfolge. Als Beispiele können die folgenden Prinzipien genannt werden: tragende vor aufliegenden Elementen, innenliegende vor äußeren Elementen und erforderliche Wartezeiten.
- Soft-Constraints beschreiben die zweckmäßigen Abhängigkeiten und bilden die gewünschte Reihenfolge der zugehörigen Bauaufgaben ab, z.B. sind die Malerarbeiten in einem Raum aus zweckmäßigen Gründen vor den Bodenbelagarbeiten auszuführen. Diese angestrebte Reihenfolge wird in diesem Beitrag als ein Soft-Constraint betrachtet.

2.2 Globale Abhängigkeiten zwischen mehreren Vorgängen (Globale Constraints).

"Globale Constraints entsprechen einer Konjunktion von Constraints. Man betrachtet diese Constraints zusammen und kann daher stärkere Lösungsalgorithmen einsetzen" (HOFSTEDT, WOLF 2007, S. 94). Diese Art der Constraints bezieht sich auf mehrere Vorgänge zusammen - wenn z.B. die Vorgänge eines Gewerkes nur innerhalb einer bestimmten vertraglichen Zeitperiode ausgeführt werden können,

wobei die einzuhaltenden Termine vertraglich festgelegt werden müssen (PROPOROWITZ, MALPRICHT, WOTSCHKE 2008, S. 155) oder wenn eine Liste von Vorgängen - wegen Raumbelagungsbeschränkungen o.ä. - gleichzeitig, nicht ausgeführt werden darf. Ein Beispiel ist, dass Malerarbeiten und Bodenbelagsarbeiten nicht gleichzeitig in einem Raum durchgeführt werden dürfen. Dafür können globale Constraints eingesetzt werden, welche die möglichen Kombinationen "Malerarbeiten vor Bodenbelagsarbeiten" sowie auch "Bodenbelagsarbeiten vor Malerarbeiten" erlauben, aber deren gleichzeitige Ausführung verhindern.

3 Constraint-basierte Ansätze im Baubetrieb

Bei COPs (Constraint Optimization Problems) kann man die optimale Lösung durch den Einsatz von Constraint-Lösern finden. Das ist im Allgemeinen nur mit einem sehr großen Rechenaufwand möglich. In KÖNIG, BEIßERT und BARGSTÄDT (2007) basiert daher die Lösung auf einer ereignis-diskreten Simulation. In jedem Simulationszeitschritt werden die Arbeitsschritte bestimmt, deren Hard-Constraints erfüllt sind und die damit ausführbar sind. Eine Auswahl der tatsächlich auszuführenden Schritte erfolgt nach einer Strategie, die in den Soft-Constraints definiert ist.

In BEIßERT, KÖNIG und BARGSTÄDT (2008) wird die Strategie "Verschmutzung vermeiden" in Ausbauprozessen als Beispiel für die Optimierung von Ausbaustrategien betrachtet. Die Arbeitsschritte werden über Soft-Constraints gemäß der Verschmutzung gewichtet. Durch eine lokale Suche werden in jedem Simulationsschritt aus den möglichen Arbeitsschritten (Hard-Constraints) diejenigen für die Ausführung ausgewählt, die die höchste Verschmutzung hervorrufen.

Im Rahmen unseres Beitrages werden die Reihenfolgen der Bauaufgaben in Form von Hard- und Soft-Constraints modelliert. Das gesamte resultierende COP wird mit dem Constraintssolver firstCS gelöst, wobei die ermittelte Lösung alle Hard-Constraints gewährleistet und eine maximale Anzahl von Soft Constraints erfüllt.

4 Problem der Optimierung von Reihenfolgen

Das Problem der Optimierung von Reihenfolgen in Bauabläufen tritt auf, wenn zeitparallel ausführbare Vorgänge nacheinander ablaufen sollen, weil u.U. dieselbe Kolonne die entsprechenden Aufgaben ausführen soll oder andere Kapazitätsbeschränkungen oder Raumbelagungsbeschränkungen existieren. Verschiedene Ausführungsstrategien der Bauproduktion stehen in vielen Fällen im Widerspruch zueinander. Das bedeutet, dass für jede Ausführungsstrategie ein Optimum berechnet werden kann (BEIßERT, KÖNIG, BARGSTÄDT 2008) und daher verschiedene Optima für verschiedene Ziele ermittelt werden können. Gesucht wird jedoch ein Pareto-Optimum für alle Ziele zusammen, was mathematisch sehr rechenaufwändig sein kann (BURKE, KENDALL 2005, S. 288). Aus diesem Grund wird für den praktischen Einsatz eine gewichtete Summe der Zielkriterien untersucht, die bei Einsatz einer geeigneten Suchheuristik (siehe Kap. 7.3), die mit vertretbarem Rechenaufwand Lösungen ermittele.

5 Modellierung des Ablaufproblems

Das Planungsproblem sei durch eine endliche Menge von Bauaufgaben und Bauplätzen (Ebenen/Räume) charakterisiert, die einer endlichen Menge von Hard-Constraints und Soft-Constraints unterliegen. Auf der Basis der festgelegten Planungszeiträume der unterschiedlichen Gewerke und dem insgesamt zur Verfügung stehenden Zeitraum ist eine optimale Lösung gesucht.

Jede Bauaufgabe ist durch die folgenden Größen charakterisiert: Startzeit, Dauer und Ende. Die Start- und Endzeit sind variabel und sollen im Planungszeitraum liegen. Es gilt auch $\text{Start} + \text{Dauer} = \text{Ende}$. Die Dauer jeder Aufgabe ist fest und wird durch Abschätzung des Aufwandes ermittelt.

Das CSP (Constraint Satisfaction Problem) besteht aus den folgenden drei Elementen: den Variablen, ihren Wertebereichen und den Constraints.

- Die Variablen sind die Start- und Endzeiten der Bauaufgaben,
- Wertebereiche sind mögliche Intervalle der Start- und Endpunkte der Bauaufgaben und
- Constraints beschreiben die Beziehungen zwischen den Bauaufgaben.

Das Optimierungsproblem besteht nun darin, die Bauabläufe so zu planen, dass alle Hard-Constraints erfüllt sind und die maximale Erfüllung der Soft-Constraints unter Einhaltung der zur Verfügung stehenden Zeiträume der unterschiedlichen Gewerke sowie der gesamten Ablaufzeit garantiert wird.

6 Lösungsansatz

In diesem Beitrag soll eine Näherungslösung vorgestellt werden, die mit vertretbarem Rechenaufwand alle Hard-Constraints erfüllt und die gesamten Soft-Constraints so gut wie möglich erfüllt.

Die vier bekannten Verknüpfungsmöglichkeiten der Netzplantechnik sind in dieser Klasse dadurch integriert, dass man die Start- oder Endevariable eines Vorgangs (x) mit der Start- oder Endevariable eines Vorgangs (y) miteinander in einem Constraint verknüpft. So wird beispielsweise die Bedingung, dass y nach x auszuführen ist, durch die Ungleichung $\text{Ende}(x) \leq \text{Start}(y)$ dargestellt.

Alle Soft-Constraints werden durch Differenzen zwischen den Start-/Endzeiten zweier Vorgänge x und y modelliert: $\text{Start}(y) - \text{Ende}(x)$. Der Erfüllungsgrad wird durch normierte Bewertungen b folgendermaßen definiert: Ist die Differenz $\text{Start}(y) - \text{Ende}(x) > 0$, dann ist die Bewertung $b = 1$, ist sie kleiner 0, dann ist die Bewertung $b = -1$, ansonsten sei $b = 0$. Für die Modellierung von Soft-Constraints werden dafür die Constraint-Klassen "DistanceSign" eingesetzt.

Diese Bewertung bildet die Basis für die Optimierung der Reihenfolgen der Vorgänge. Die Reihenfolge, die die maximale Erfüllung der Soft-Constraints erreicht, ist die optimale Ausführungsvariante. Die Zielfunktion, die es zu optimieren gilt, ist die Summe der normierten Bewertungen aller Soft-Constraints.

7 Beispiel

Das folgende Beispiel betrachtet das Ablaufproblem in einem sechsstöckigen Gebäude. Zunächst werden die in jeder Etage auszuführenden Bauaufgaben und deren Abhängigkeiten analysiert.

7.1 Konstruktion der Etagen im Beispiel

Im folgenden Beispiel werden mehrere Ablaufreihenfolgen für die in Abbildung 1 vorgestellte Konstruktion innerhalb einer Etage gezeigt.

Die Ablaufanalyse dient der Beantwortung der Frage, ob ein Vorgang vor dem anderen aus zweckmäßigen oder zwingenden Gründen auszuführen ist. Die Grundlage dieser Analyse bilden die Abhängigkeitsbeziehungen zwischen den Vorgängen. Die Analyse kann in die lokale und globale Ablaufanalyse unterteilt werden. Die lokale Ablaufanalyse betrifft die am selben Ort auszuführenden Vorgänge. Diese Analyse wird hier für die Ausbaurbeiten in einem Raum am Beispiel einer Trennwandkonstruktion (siehe Abb. 1) durchgeführt.



Abbildung 1: Ablaufvarianten innerhalb eines Raumes (SOMMER 1994, S. 92)

Im Beispiel ist eine Trennwand vom Rohboden bis zur Rohdecke zu erstellen. In dieser Variante kann entweder mit der Trennwand oder mit der Deckeninstallation begonnen werden. Danach ist die abgehängte Decke oder die komplette Fußbodeninstallation durchzuführen. Das ergibt insgesamt vier mögliche Ablaufvarianten (aus Sicht der Autoren), wobei alle Räume einer Etage entsprechend einer gemeinsamen Variante gebaut werden.

Die folgenden fünf Hard-Constraints sind innerhalb der Etage zu betrachten: Aufgabe 5 vor Aufgabe 2, Aufgabe 5 vor Aufgabe 3, Aufgabe 5 vor Aufgabe 4, da die Trennwand vor der abgehängten Decke, dem Doppelboden sowie der Installation des Fußbodens auszuführen ist. Aufgabe 1 vor Aufgabe 2, Aufgabe 4 vor Aufgabe 3, da innenliegende Elemente vor äußeren Elementen auszuführen sind.

Die beiden Soft-Constraints sind: Aufgabe 5 vor Aufgabe 1, damit die Montagewände genau ausgerichtet werden und die Installation darauf abgestellt werden kann. Aufgabe 2 vor Aufgabe 3, da die oberen Arbeiten vor den unteren Arbeiten aus mehreren Gründen besser auszuführen sind.

In dem Beispiel empfiehlt der Autor, die Variante A auszuwählen, da alle zweckmäßigen und auch zwingenden Abhängigkeiten eingehalten sind. Es ist aber auch der Ablauf C möglich. Die Abläufe B und D sind grundsätzlich möglich, da diese die zwingenden Bedingungen (Hard-Constraints) erfüllen; sie sind jedoch weniger geeignet, da zweckmäßige Bedingungen (Soft-Constraints) ignoriert werden.

In Tabelle 1 werden die möglichen Ablaufvarianten mit der Zahl der erfüllten Soft-Constraints und deren erreichten Zielfunktionswert innerhalb einer Etage nach der Lösung mit Hilfe der Constraint-Bibliothek firstCS dargestellt.

Variante	Reihenfolgen					Soft-Constraints	Zielfunktionswert
A	5	1	2	4	3	2	2
B	5	1	4	3	2	1	0
C	1	5	2	4	3	1	0
D	1	5	4	3	2	0	-2
E	5	4	3	1	2	1	0

Tabelle 1: Mögliche Ablaufvarianten innerhalb einer Etage

Der Tabelle ist zu entnehmen, dass die Variante A die optimale Variante ist. Zusätzlich zu den vom Autor in Abbildung 1 dargestellten vier Varianten ist eine weitere fünfte Variante (E) möglich. Diese Variante ist zwar keine optimale Variante könnte aber bei einer Planänderung nützlich sein.

7.2 Das gesamte Gebäude

Das erweiterte Beispiel besteht aus sechs Stockwerken, die dieselbe Konstruktion wie Abbildung 1 haben. Zur Festlegung der Constraints wird der allgemeine Ablauf der Gewerke analysiert. Diese Analyse liefert die sinnvollen und zwingenden Fertigungsrichtungen der einzelnen Gewerke. In diesem Beispiel werden die Gewerke, wo möglich, die unteren Geschosse zuerst bearbeiten. Es könnte aber auch eine Fertigungsrichtung von oben nach unten oder eine gemischte Fertigungsrichtung eingegeben werden, je nach Wunsch der beteiligten Gewerke.

Innerhalb jeder Etage sollen sowohl die Hard-Constraints als auch die Soft-Constraints (Kap. 7.1) berücksichtigt werden. Weiterhin kommen zusätzliche Soft-Constraints hinzu, die die gewünschte Reihenfolge der gleichen Bauaufgaben in unterschiedlichen Etagen beschreiben, z.B. Trennwand im Erdgeschoss vor der Trennwand im 1. Obergeschoss, Trennwand im 1. Obergeschoss vor der Trennwand im 2. Obergeschoss, usw. Das ergibt für die sechs sich wiederholenden Aufgaben weitere fünf Soft-Constraints, die deren gewünschte Reihenfolge beschreiben. Zusammen sind dies dann zusätzliche 25 Soft-Constraints. Dabei wird folgender Zeitbedarf pro Etage angenommen: 2 Tage für die Trennwände, 4 Tage für die Deckeninstallation, 3 Tage für die abgehängte Decke, 4 Tage für die Installation des Fußbodens und 2 Tage für den Bodenbelag Sofern alle Soft-Constraints eingehalten werden, wird eine lange Gesamtdauer von 45 Tage benötigt. Ist die zur Verfügung stehende Ausführungsdauer kürzer als 45 Tage, so sind die Reihenfolgen zu berechnen, die eine maximale Erfüllung von Soft-Constraints gewährleisten. Die folgenden zwei Constraints werden dabei als globale Constraints eingesetzt:

- *Zeitlich überlappungsfreie Platzierung von Arbeitsgängen auf einer Resource*: sowohl für die Arbeitsressourcen als auch die Arbeitsplätze.
- *Paarweise Verschiedenheit (AllDifferent)*: verhindert, dass die zugehörigen Variablen denselben Wert annehmen können. Z.B. die Abbrucharbeiten der Deckeninstallation dürfen nur in einer Etage gleichzeitig stattfinden.

7.3 Lösung des Constraint-Optimierungsproblems mittels problemangepasster Suche

Zur Lösung des Constraint-Optimierungsproblems wurde die Branch&Bound-Methode so umgesetzt, dass beginnend mit der Suche nach einer initialen Lösung, durch Beschränkung des Zielfunktionswerts nach immer besseren Lösungen im verbleibenden Suchraum gesucht wird, bis letztlich eine optimale Lösung gefunden wurde, d. h. wenn eine Lösung mit Zielfunktionswert v gefunden wurde, wird dies durch Beschränkung der Zielfunktion g mittels $g > v$ erreicht. Der Einsatz einer naiven Tiefensuche mit chronologischem Rücksetzen beim Auftreten einer Inkonsistenz, das jeder Startzeit-Variablen einen Wert zuordnet, ist – wie unsere Versuche gezeigt haben – nicht zielführend: Die Optimierung war auch nach über einer Stunde Rechenzeit erfolglos. Ein Grund dafür ist, dass es viele Wertekombinationen gibt, welche weder die Reihenfolge der Vorgänge noch die Bewertung der Soft-Constraints (Vorzeichen von Differenzen) verändern, was bei der Suche einen sogenannten "Thrashing"-Effekt ("auf der Stelle treten") zur Folge hat. Aus diesem Grund wurde zur Optimierung eine chronologische Tiefensuche eingesetzt, die zwei zeitlich anzuordnende Vorgänge, die noch ungeordnet sind, auswählt und ordnet, so dass durch die beiden Alternativen "A vor B" oder "B vor A" ein binärer Suchbaum aufgespannt wird, der durch Ausschluss widersprüchlicher Teilordnungen beschnitten wird, so dass maximal $n!$ lineare Ordnungen bei n Vorgängen betrachtet werden, d.h. die Suche terminiert nach endlich vielen Schritten, sobald die Vorgänge linear geordnet sind und damit alle Bewertungen der Soft-Constraints eindeutig sind. Eine Belegung der Startzeit-Variablen mit Werten ist nicht erforderlich – ganz im Gegenteil: die Größe der Wertebereiche gibt Hinweise auf zeitliche Puffer: ist der Start eines Vorgangs durch die Anordnung bestimmt, so liegt dieser Vorgang auf einem "kritischen Pfad", jede zeitliche Abweichung verletzt entweder ein Hard-Constraint oder führt zumindest zu einer suboptimalen Lösung.

7.4 Ergebnisse

Für eine gegebene Zeitdauer wird das Optimierungsproblem gelöst. Dazu werden Ablaufreihenfolgen ermittelt, die die Einhaltung einer maximalen Anzahl von Soft-Constraints erlauben. Für eine maximale Ablaufzeit von 35 Tagen ergeben sich z.B. 12 optimierte Varianten mit derselben Anzahl erfüllter Soft-Constraints. Aus einer möglichen Variante sind in Tabelle 2 die Intervalle der Startzeiten der Aufgaben Trennwände im gesamten Gebäude dargestellt. Diese Ergebnisse wurden auf einem Windows-PC (2GHz CPU, 2GByte RAM) in wenigen Sekunden berechnet.

Baufgabe	Etage	Startzeiten(Tagen)
Trennwände	EG	0
Trennwände	1 OG	2..4
Trennwände	2 OG	6..8
Trennwände	3 OG	8..12
Trennwände	4 OG	10..16
Trennwände	5 OG	4..6

Tabelle 2: Startzeiten der Aufgaben Trennwände

8 Zusammenfassung und Ausblick

Es wurde ein constraint-basierter Lösungsansatz zur Bauablaufplanung im Hochbau vorgestellt. Zukünftige Arbeiten zur Erweiterung des Modells umfassen die Definition von Constraints für die unterschiedlichen Ressourcenarten (exklusiv, kumulativ, etc.), sowie die Entwicklung von Methoden zur Gewichtung der Soft-Constraints auf der Basis einer Nutzwertanalyse. Ferner muss das gesamte Konzept anhand eines realen Projekts validiert werden.

Literatur

- BEIßERT, Ulrike; KÖNIG, Markus; BARGSTÄDT, Hans-Joachim: Simulation von Ausführungsstrategien mit Hilfe von Soft-Constraints und Lokaler Suche. In: *Advances in Simulation for Production and Logistics Applications*. Hrsg.: RABE, Markus. Stuttgart: Fraunhofer IRB Verlag, 2008, ASIM 2008, S. 141-150.
- BRANDENBERGER, Jürg ; RUOSCH, Ernst: *Ablaufplanung im Bauwesen*, Dietikon: Baufachverlag AG, 1993.
- BURKE, Edmund K.; KENDALL, Graham : *Search Methodologies*, New York: Springer Verlag, 2005.
- HOFSTEDT, Petra; WOLF, Armin: *Einführung in die Constraint-Programmierung*, Berlin: Springer-Verlag, 2007.
- HUHNT, Wolfgang; ENGE, Felix : *Simulationsmodelle für die Ausführung von Bauleistungen*" In: 1. IBW-Workshop Simulation in der Bauwirtschaft. Hrsg.: FRANZ, Volkard. Kassel: kassel university press, 2007, S. 29-50
- KÖNIG, Markus; BEIßERT, Ulrike; BARGSTÄDT, Hans-Joachim: Ereignis-diskrete Simulation von Trockenbauarbeiten - Konzept, Implementierung und Anwendung. In: 1. IBW-Workshop Simulation in der Bauwirtschaft. Hrsg.: FRANZ, Volkard. Kassel: kassel university press, 2007, S. 15-27.
- NÄGEL, Ulrich; GÖTTING, Kurt; HÄNEL, Klaus; WAGNER, Christoph: *Ausbauprozesse*. Berlin : VEB Verlag für Bauwesen, 1990.
- PROPOROWITZ, Armin; MALPRICHT, Wolfgang; WOTSCHKE, Michael: *Baubetrieb-Bauwirtschaft*. Leipzig: Fachbuchverlag im Carl Hanser Verlag, 2008.
- ROSSI, Francesca; VAN BEEK, Peter; WALSH, Toby: *Handbook of Constraint Programming*, Amsterdam : 1. Aufl. Elsevier, 2006.
- SOMMER, Hans.: *Projektmanagement im Hochbau*, Berlin, Heidelberg: Springer-Verlag, 2. Auflage 1994.
- WOLF, Armin: Object-Oriented Constraint Programming in Java Using the Library firstes. In: 20th Workshop on Logic Programming. Wien: Technische Universität, 2006, S. 21-32.