

Deklarative Modellierung und effiziente Optimierung

(MOC 2011)

Ulrich Geske¹, Armin Wolf²

¹Universität Potsdam, ²Fraunhofer FIRST, Berlin
Ulrich.Geske@uni-potsdam.de, Armin.Wolf@first.fraunhofer.de

Vorwort zum Workshop

Für die Unterstützung bei der Organisation und Durchführung des nunmehr vierten Workshops dieser Serie danken wir dem Programmkomitee, das insbesondere durch die Begutachtung der eingereichten Beiträge und die Unterstützung der Autoren durch Hinweise zu ihren Arbeiten zum Gelingen beigetragen hat: Christoph Beierle, FernUniversität Hagen; Hans-Joachim Goltz, Fraunhofer FIRST, Berlin; Petra Hofstedt, Brandenburgische Technische Universität Cottbus; Walter Hower, Hochschule Albstadt-Sigmaringen; Ulrich John, SIR Dr. John UG, Berlin; Frank Raiser, Universität Ulm; Georg Ringwelski, Hochschule Zittau/Görlitz; Dietmar Seipel, Universität Würzburg.

Die aktuelle Entwicklung tendiert immer mehr dahin, auch für große und komplexe Probleme in unterschiedlichsten Lebensbereichen in kürzester Zeit eine Lösung und häufig sogar eine nahezu optimale Lösung finden zu müssen. Die Programmierung kombinatorisch komplexer Probleme, basierend auf der Grundlage von Constraint-Techniken, weist inzwischen eine Anwendungsreife auf, dass sie zur Modellierung und Lösung vielfältiger Optimierungsprobleme, bei denen eine Vielzahl meist unterschiedlicher Bedingungen zu berücksichtigen sind, einsetzbar ist. Mit diesem Verfahren bearbeitete Anwendungsbereiche von Simulations- und Optimierungs-Verfahren sind vielfältig. Sie liegen z.B. im Bereich der Planung und Optimierung der Reihenfolge von Operationen, der Personalplanung (Dienstpläne unter Beachtung unterschiedlicher Qualifikationen), aber auch bei Konfiguration und Wartung medizintechnischer Geräte und der Energieeinsparung (z.B. Reduzierung des Energieverbrauchs von Anlagen als ein Optimierungskriterium) liegen. Für Unternehmen ist es häufig wichtig, dass innerhalb kürzester Zeit eine akzeptable Lösung gefunden wird, um wettbewerbsfähig zu sein. Dabei muss auf veränderte Randbedingungen reagiert werden können, um die Lösung anzupassen. Für die Simulationen von "was-wäre-wenn"-Szenarien sind Konsequenzen aus Entscheidungen unmittelbar aufzuzeigen, um bestmögliche Unterstützung der Anwender zu bieten.

Zur Optimierung eingesetzte Methoden weisen eine große Breite auf. Neben der Weiterentwicklung von mathematischen Verfahren des Operations Research und der Constraint-Programmierung basieren erfolgreiche moderne Verfahren und Programmierwerkzeuge auf Methoden des Soft Computing (wie Genetische

Algorithmen), auf heuristischen (wie Simulated Annealing), wissensbasierten (wie Discrepancy Search) und mathematischen (wie Operations Research) Verfahren. Das besondere Interesse des Workshops ist auf die Untersuchung, Weiterentwicklung und Anwendung der Technik der Constraint-Löser gerichtet. Constraint-Verfahren zeichnet unter den genannten Verfahren der Vorteil der größtmöglichen Transparenz der Lösungsfindung aus. Die Nachvollziehbarkeit der automatisch gefundenen Lösung, ist für den Anwender häufig ein wesentliches Auswahlkriterium. Sofern Nachvollziehbarkeit bei komplexen Problemen überhaupt vorstellbar ist, bieten Constraint-Verfahren auf Grund ihrer problembezogenen Modellierung hierfür aber den bestmöglichen Ansatz. Ein weiteres wesentliches Merkmal dieses Verfahrens ist die Möglichkeit einer inkrementellen, anwendergesteuerten Lösungsverbesserung.

Von den unterschiedlichen Paradigmen der Constraint-Programmierung ist das der *Endlichen Domänen* für viele industrielle und im weitesten Sinne verwaltungstechnische Anwendungsfälle besonders geeignet. Den das Problem kennzeichnenden Variablen sind initial ganzzahlige und endliche Wertebereiche zugeordnet, die dem problemspezifischen Wertevorrat entsprechen. Die Variablen stehen im Allgemeinen untereinander in gewissen Relationen. Diese Relationen können einfache arithmetische Vergleiche zwischen zwei Variablen (einfache Constraints genannt) oder komplexere Bedingungen (globale Constraints) sein. Sie drücken Bedingungen aus, die eine Lösung des Problems erfüllen muss. Vor der Durchmusterung des Suchraums, der durch die Variablen und ihre Werte aufgespannt wird, werden durch geeignete Konsistenztechniken Werte aus den Wertebereichen der Variablen entfernt, die garantiert zu keiner Lösung beitragen können. Die Entfernung von Werten einer Variablen beeinflusst über die Constraints die Wertebereiche von anderer Variablen. Die Folgen des Entferns von Werten einer Variablen werden (rekursiv) über alle Variablen propagiert. Der Effekt dieser Propagation besteht in einer, im Allgemeinen erheblichen, Reduktion des Suchraums, so dass die nachfolgende Suche effizienter ablaufen kann. Die Effizienz wird dadurch verstärkt, dass während der Suche ebenfalls Propagation stattfindet.

Suchverfahren sind neben Konsistenztechniken ein weiterer wesentlicher und im Allgemeinen unverzichtbarer Bestandteil von Constraint-Systemen. Suche bedeutet, nach einem gewissen Verfahren eine Variable auszuwählen und einen (willkürlichen) Wert aus dem nach der Propagation verbliebenen Wertebereich der Variablen als endgültigen Wert dieser Variablen festzulegen. Bevor die analoge Aktion für eine weitere Variable geschieht, wird diese Festlegung ebenfalls propagiert und reduziert somit erneut den Suchraum. Da die willkürliche Werteauswahl in einer Sackgasse enden kann bzw. der Umfang der angestrebten Suchraumreduzierung beispielsweise von der Variablen- und Werteauswahl abhängt, sollten unterschiedliche Suchverfahren verfügbar oder spezifizierbar sein.

Das beschriebene Verfahren der Constraint-Verfahren bedingt angepasste Systeme, als Constraint-Solver bezeichnet, in Form von Compilern bzw. Bibliotheken, da der Propagationsschritt innerhalb der Programmabarbeitung bedeutet, dass die das Problem kennzeichnenden Constraints ausgeführt werden müssen, bevor die enthaltenen

Variablen einen festen Wert besitzen. Derartige Constraint-Solver sind auf unterschiedlicher Basis verfügbar. Um die Nutzung backtracking-basierter Suche durch den Systemansatz zu unterstützen sind constraint-logische Systeme geeignet, Java-basierte Constraint-Solver stellen die Verbindung zur klassischen Programmierung her.

Die Effizienz der constraint-basierten Verfahren liegt nicht nur in der Verkleinerung des Suchraums begründet, sondern auch in der Tatsache, dass Sackgassen in der Lösungsfindung frühzeitig erkannt werden. Während klassische Generate-and-Test-Verfahren vor dem Test der Bedingungen die Belegung aller beteiligten Variablen mit Werten erfordern, ist in der Constraint-Programmierung mit endlichen Wertebereichen eine Sackgasse unabhängig von der Belegung anderer Variablen einfach dadurch erkennbar, dass im Propagation-Schritt der Wertebereiche irgendeiner Variablen leer wird.

Die Workshop-Reihe *Deklarative Modellierung und effiziente Optimierung* zielt insbesondere auf die Anwendung und Weiterentwicklung von Constraint-Technologien sowie auf die Verbindung bzw. Gegenüberstellung dieses Paradigmas mit anderen Optimierungsverfahren. Konkrete Themen, die für den Workshop eine Rolle spielen sind deshalb einerseits die constraint-basierte Modellierung und Lösung von Anwendungsproblemen, u.a. für die Werkstatt- und Materialbedarfsplanung, das Flottenmanagement und die Fahrplanoptimierung, Personal- und Stundenplanung, Variantenkonfiguration technischer Anlagen aber auch deren Software, die Optimierung verteilter Berechnungs- oder Geschäftsprozesse sowie die Lastverteilung, als auch andererseits die Implementierung von Constraint-Modellen, u.a. durch die Verwendung unterschiedlicher Constraint-Systeme, die Erweiterung von Konsistenztechniken, effiziente Suchverfahren, Nachweis der Optimalität, Visualisierung des Lösungsprozesse, inkrementelle Lösungsanpassung u.a. in Verbindung mit anderen Optimierungstechniken. Die für diesen Workshop akzeptierten Arbeiten demonstrieren einige der unterschiedlichen Aspekte des Workshop-Themas.

In ihrer Arbeit „Optimierte Ablaufpläne für Ressourcenverbraucher bei zeitvariablen Ressourcentarifen“ zeigen Walter Weinger und Armin Wolf detailliert die Art und Weise der Herangehensweise bei der Problemspezifikation unter Anwendung der Constraint-Technologie. Das Ziel der Problembehandlung ist es, bei zeitvariablen Stromtarifen minimale mögliche Verbrauchskosten durch eine geeignete zeitliche Verteilung der Nutzung stromverbrauchender Geräte zu erreichen. Das erzielbare Einsparpotential ist durch die Differenz von minimal und maximal möglichen Verbrauchskosten anzugeben. Die Autoren untersuchen, wie sich unterschiedliche Modellierungsarten auf die Qualität des Ergebnisses und die zur Lösungsfindung benötigte Zeit auswirken.

Saskia Sandow beschreibt in ihrem Beitrag “Ressourcenplanung unter Nutzung der Java-Constraint-Bibliothek firstCS“ eine mehrstufige Systemkonfiguration für die Beschreibung und Bearbeitung von Ressourcenplanungsproblemen. Das spezielle Anwenderproblem kann mit Hilfe einer XML-Datei deklarativ spezifiziert werden. Das universell einsetzbare Ressourcenplanungssystem ist mittels einer Konfigurations-Datei für die effiziente Bearbeitung des speziellen Problems, insbesondere auch bezüglich der

Suchstrategie, anpassbar. Die Zielrichtung der Arbeit besteht darin, dem Anwender die Möglichkeit zu geben, problemangepasste Strategien auf einem hohen sprachlichen Niveau zu spezifizieren.

Christoph Beierle und Gabriele Kern-Isberner beschreiben in der Arbeit „On the Computation of Ranking Functions for Default Rules – A Challenge for Constraint Programming“ eine Anwendung der Constraint-Programmierung zur Bewertung der Interpretationen logischer Regeln. Das Auffinden der Vorzugsbedeutung einer Implikation hat erheblichen Einfluss auf die Lösung eines Problems. Die Autoren verweisen auf die hohe deklarative Natur der Constraint-Programmierung, die es ermöglicht, in einfacher Weise Änderungen in der Spezifikation der Lösungsstrategie vorzunehmen.

Steve Dillan, Anna Prenzel und Georg Ringwelski diskutieren in „Evaluating a Practical Decision Support System from Usability Perspective“ das Problem der Schnittstelle von Softwareentwickler und Anwender. Auch wenn ein Verfahren bessere Lösungen als andere Verfahren liefert, muss es nicht zwangsläufig vom Anwender akzeptiert werden. Die Autoren schlagen zur Untersuchung der für Softwarefirmen und Anwender gleichermaßen essentiellen Zusammenhänge eine Fallstudie mit unterschiedlichen Szenarien vor. Das Testbett für die Fallstudie ist ein von ihnen entwickeltes constraint-basiertes Optimierungsverfahren zur Routenplanung.

Berlin, im Juli 2011

Ulrich Geske und Armin Wolf