

Diplomthema: Lösung komplexer "Pickup and Delivery" Probleme unter Einbeziehung moderner Constrainttechniken

vorgelegt von: Mathias Lühr

Die Aufgabe

Für die Tourenplanung innerhalb der Entsorgungs- und Baubranche sollen moderne Constrainttechniken analysiert und eine Softwarebibliothek entwickelt werden. Hierzu sollte der Stand der Forschung auf den Gebieten der "Constraint-Satisfaction" und von "Constraint Systemen" einfließen und bei der Analyse der Fahrzeugplanung hinsichtlich der Anforderungen an die zu schaffende Softwarebibliothek beachtet werden.

Für die Arbeit sollten geeignete Constraint-Solver-Bibliotheken hinsichtlich des Fahrzeugroutens evaluiert und ein zweckmäßiger Constraint-Solver ausgewählt werden. Unter Einbeziehung des gewählten Solvers stand es zur Aufgabe, eine Softwarebibliothek zu entwerfen und zu implementieren, die gegebene "Pickup and Delivery" Probleme löst. Die Implementierung sollte in C/C++ oder in Delphi (Objektpascal) durchgeführt werden. Dabei war zu beachten, dass die entwickelte Bibliothek von Delphi-Programmen aus nutzbar und auf dem Betriebssystem Windows NT 4 lauffähig sein musste.

Betreuer der Arbeit waren von Seiten der Hochschule Herr Prof. Dr.-Ing. Jochen Heinsohn und seitens der Firma IVU Traffic Technologies AG, an der die Arbeit erstellt wurde, Herr Dipl.-Ing. (FH) Torsten Storrer.

Die Arbeit

"Pickup and Delivery" Probleme sind Routingprobleme, bei denen in irgendeiner Form ein Fahrzeug an einer Position beladen und an einer anderen Position entladen werden muss.

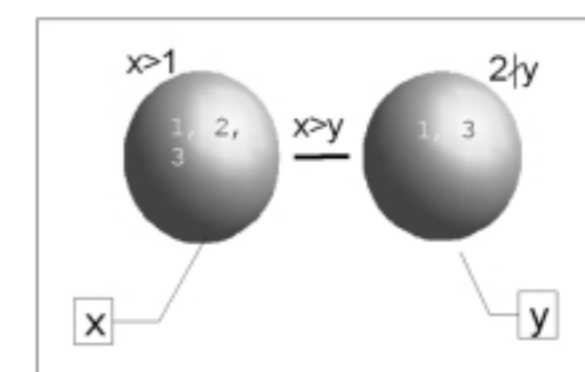
Dazu zählen beispielsweise die Verplanung einer Entsorgungsfahrzeugflotte oder Fahrzeugplanung von Kurierdiensten. Eine nahe Verwandtschaft dieser Problemklassen und dem Traveling Salesman Problem, eines Handelsreisenden, der die kürzeste Route zwischen mehreren Anlaufstationen finden muss, kann bewiesen werden. Das Traveling Salesman Problem liegt in der Klasse NP, der Menge aller Probleme, die mit Hilfe nichtdeterministischer Algorithmen in polynomialer Zeit gelöst werden können. Weiterhin wird angenommen, dass für das Traveling Salesman Problem kein deterministischer polynomialer Algorithmus existiert.

Die Schwierigkeit solcher Probleme ist die, dass die heute bekannten Algorithmen sie mit einer Größenordnung von $O(n!)$ nicht in akzeptabler Zeit lösen können. Die Komplexität ergibt sich aus der vollständigen Enumeration aller möglichen Lösungen, um schlussendlich die beste auszugeben. Diese kann eingeschränkt werden, indem Randbedingungen berücksichtigt werden und so den Suchraum einschränken.

1. Constraints

Constraints sind Beschränkungen, wie $x > y$ oder $x > 1$. Außerdem wirkt ein Constraint immer auf eine oder mehrere Constraintvariablen und schränkt deren Wertebereich ein. Der Wertebereich einer solchen Variablen wird auch Domain genannt und ist im Allgemeinen endlich.

Mehrere Variablen, auf die mehrere Constraints wirken, können ein Constraintnetz bilden. Ein solches Netz liegt genau dann vor, wenn Constraints vorhanden sind, die auf maximal zwei Variablen wirken. Ein Constraintnetz lässt sich wie in der nebenstehenden Abbildung darstellen. Dabei kann sich eine Änderung in einer Domain, verbunden durch ein Constraint, auf eine andere Domain auswirken. Die Fortpflanzung der Beschränkungen heißt ConstraintPropagierung.



2. Alternativen zur Constraintprogrammierung

Für einige Probleme lassen sich Alternativen zur Constraintprogrammierung finden.

Die Lineare Programmierung kommt aus dem Bereich der Unternehmensforschung und erzeugt ein mathematisches Modell, welches z.B. mit Hilfe des Simplexverfahrens gelöst werden kann. Dieses Verfahren basiert auf der Gauß-Jordan-Eliminationsmethode und sucht zielgerichtet eine optimale Lösung.

Mit Hilfe der Simulation von komplexen Systemen können Vorhersagen auf bestimmte Ereignisse getroffen werden. Dazu wird ein System in Teilsysteme zerlegt. Eine Kombination dieser Teilsysteme kann mit einer gewissen Wahrscheinlichkeit die Lösung des Problems vorhersagen.

Evolutionäre Algorithmen bilden eine Lösung als Chromosom ab. Die Güte der Chromosomen bestimmt ihr Überleben und ihre Weiterentwicklung. Mit Hilfe von Rekombination und Mutation werden neue Lösungen erzeugt, welche eine neue und eventuell qualitativ bessere Population bilden.

Die Vorteile des Teile-und-Herrsche-Prinzips versucht die Dynamische Programmierung weiter auszubauen. Das Problem wird in mehrere Teile zerlegt und anschließend versucht, unter Einbeziehung vorheriger Ergebnisse kleinerer Teile, das Gesamtproblem zu lösen.



3. Systemauswahl

Untersucht wurden die fünf Constraintlöser ECLiPSe, CHIP, Mozart/OZ, StarFlip und die ILOG-Bibliotheken, um das optimalste System für die Lösung von Vehicle-Routing-Problemen zu finden. ECLiPSe (ECRC Logic Programming System) ist eine Erweiterung von Prolog und beherrscht den Umgang mit Arrays und Constraints. Darüber hinaus besteht die Möglichkeit, ECLiPSe-Programme mit C/C++-Code zu erweitern oder ECLiPSe-Bibliotheken von C/C++-Programmen aus zu benutzen.

CHIP (Constraint Handling in Prolog) ist ursprünglich als Erweiterung zu Prolog entstanden. CHIP ist eine ausgereifte Constraint-Bibliothek, die über umfassende Möglichkeiten verfügt, Constraints zu deklarieren. Neben einer C/C++-Schnittstelle existiert ein Prolog-Interface.

OZ ist eine Multi-Paradigmen-Sprache, die Eigenschaften der Objektorientiertheit und der funktionalen und deklarativen Programmierung beherbergt. Die Entwicklungsumgebung zu OZ heißt Mozart.

StarFlip kann mit Fuzzy-Sets und Fuzzy-Constraints umgehen. Vages Wissen und Beschränkungen finden in den Fuzzy-Constraints ihre Fortsetzung und zur Unterstützung der Optimierung wurden Tabu-Search und Genetische Algorithmen implementiert. Die ILOG-Solver-Bibliothek bietet eine breite Unterstützung für die objektorientierte Constraint-Programmierung. Alle wichtigen Mechanismen, die zur Lösung eines Vehicle-Routing-Problems notwendig sind, werden von der ILOG-Dispatcher-Bibliothek, die auf die Solver-Bibliothek aufsetzt, implementiert.

Die ILOG-Bibliotheken stellten die besten Voraussetzungen mit ihrer Ausrichtung auf "Pickup and Delivery" Probleme, einer C++-Schnittstelle und somit der benötigten Möglichkeit der DLL-Erstellung.

4. Programmierung

Im Ergebnis wurde eine DLL auf Basis der ILOG-Bibliotheken entwickelt, auf die von einem Delphi-Programm zugegriffen wird. Die DLL ist in der Lage, ein komplexes "Pickup and Delivery" Problem zu lösen und dabei eine einfache Schnittstelle zu bieten. Sie liest dazu die Probleminstanz ein, erzeugt die Beschränkungen mittels Constraints der ILOG-Bibliotheken und berechnet die Lösung.