n – Damenproblem

1) Internetrecherche des n – Damen Problems

Geschichte:

Erstmals formuliert wurde das Damenproblem von dem bayrischen Schachmeister Max Bezzel (1824-1871). In der "Berliner Schachzeitung" fragte er 1848 nach der Anzahl der möglichen Lösungen. Als erster nannte 1850 Dr. Franz Nauck in der "Leipziger Illustrierten Zeitung" die korrekte Zahl 92. Auch Carl Friedrich Gauß zeigte Interesse an dem Problem, weshalb es irrtümlich häufig auf ihn zurückgeführt wird.
Nauck verallgemeinerte die Problemstellung und fragte, auf wie viele verschiedene Arten man n Damen auf einem nxn-Schachbrett aufstellen könne.
1991 wurde von B. Bernhardsson eine explizite Lösung des N-Damenproblems für jede beliebige Brettgröße im ACM SIGART Bulletin, Vol. 2, No. 7 angegeben.
Im Jahre 1992 fanden Demirörs, Rafraf und Tanik eine Äquivalenz zwischen magischen Quadraten und Damenproblemen.
Das Damenproblem tauchte auch in The 7th Guest auf, einem Computerspiel aus den 1990er Jahren.

Problemstellung:

Das Damenproblem ist eine Denksportaufgabe: Man finde eine Stellung für n Damen auf einem Schachbrett, derart dass keine zwei Damen sich gegenseitig nach den Regeln des Schach schlagen können (die Figurenfarbe wird dabei ignoriert, und es wird angenommen, dass jede Figur jede andere angreifen könnte). Anders gesagt sollen sich keine zwei Damen die gleiche Reihe, Linie oder Diagonale teilen.

Es gilt, n nicht - dominierende Damen auf einem Brett von n x n Feldern zu positionieren. Für z.B. n=8 hat das Damenproblem 12 verschiedene Lösungen (92, wenn man die sich durch Spiegelung oder Rotation des Brettes ergebenden Lösungen mitzählt).

Lösung des Problems:

Das Damenproblem ist ein gutes Beispiel für ein einfach zu formulierendes Problem mit nicht trivialen Lösungen. Eine Reihe von Programmiertechniken sind geeignet, alle Lösungen zu erzeugen: Logische Programmierung, Genetische Algorithmen oder Rekursion.

Derartige Ansätze sind wesentlich effizienter als ein naiver Brute Force-Algorithmus, der (im 8×8 Fall) alle 64*63*62*61*60*59*58*57 (knapp 648 = 248) möglichen Positionierungen der acht Damen durchprobiert und dabei alle Stellungen ausfiltert, in denen zwei Damen sich schlagen könnten. Dieser Algorithmus erzeugt mehrfach die gleichen Lösungen, wenn Permutationen der Damen gleiche Felder besetzen.
Ein effizienterer Brute Force-Algorithmus platziert in jeder Reihe nur eine Dame und reduziert dadurch die Komplexität auf 88 = 224 mögliche Stellungen.

Logische Programmierung:

Logische Programmierung ist ein Programmierparadigma, das auf der mathematischen Logik beruht. Anders als bei der imperativen Programmierung besteht ein Logik-Programm nicht

aus einer Folge von Befehlen, sondern aus einer Menge von Axiomen, welche hier als eine reine Ansammlung von Fakten oder Annahmen zu verstehen sind. Stellt der Benutzer eines Logik-Programms eine Anfrage, so versucht der Interpreter die Lösungsaussage allein aus den Axiomen zu berechnen.

Dazu werden eine Menge von so genannten Regeln und Anweisungen, die der Syntax gemäß aufgebaut sind, zusammen mit der Information, welche Lösungsmethode vorgesehen ist, in den Programmcode eingefügt. Regelbasierte Programmiersprachen sind zugleich auch logische und deklarative Programmiersprachen und gehören zu den grundlegenden Werkzeugen der Künstliche Intelligenz.

In einem imperativen Programm wird genau beschrieben, wie und in welcher Reihenfolge ein Problem zu lösen ist. Im Gegensatz dazu wird in einem regelbasierten Programm idealerweise nur beschrieben, was gilt. Das "wie" ist bereits durch die Lösungsmethode vorgegeben. Die Lösung wird aus den vorhandenen Regeln hergeleitet. Meistens wird schon nur eine Menge von Regeln als "das Programm" bezeichnet, wenn klar ist, welche Lösungsmethode dazugehört: Nämlich die (einzige) in der vom regelbasierten System bereit gestellten Inferenzmaschine verwirklichte. Strenggenommen ist ein Satz von Regeln nur ein Programmtext. Diese strenge Unterscheidung ist allerdings nur selten nötig.

Genetische Algorithmen:

Genetische Algorithmen (GA) sind Algorithmen, die eine Lösung zu einem nicht analytisch lösbaren Problem finden, indem sie "Lösungsvorschläge" solange verändern und miteinander kombinieren, bis einer dieser Vorschläge den gestellten Anforderungen entspricht.

Genauer sind GA heuristische Optimierungsverfahren und gehören zu den Evolutionären Algorithmen. Sie werden vor allem für Probleme eingesetzt, für die keine geschlossene Lösung vorliegt und stehen in Konkurrenz zu klassischen Suchstrategien wie dem A*-Algorithmus, der Tabu-Suche oder dem Gradientenabstiegsverfahren.

Im Gegensatz zur Genetischen Programmierung ist das Verfahren der Genetischen Algorithmen recht unflexibel. Es werden im Normalfall lediglich die Parameter einer Gleichung, Formel oder eines in anderer Form vorgegebenen strukturierten Lösungsansatzes optimiert.

Die Grundidee ist, ähnlich der biologischen Evolution, eine Anzahl Lösungskandidaten (Individuen) zufällig zu erzeugen und diejenigen auszuwählen, die einem bestimmten Gütekriterium am besten entsprechen. Deren Eigenschaften (Parameterwerte) werden dann leicht verändert und miteinander kombiniert, um eine neue Lösungskandidaten (eine neue Generation) zu erzeugen.

Der typische GA umfasst die folgenden Schritte:
Initialisierung: Erzeugen (engl. "generate") einer ausreichend großen Menge unterschiedlicher "Individuen" (Lösungskandidaten).
Evaluation: Für jeden einzelnen Lösungskandidat wird anhand einer Zielfunktion (auch Fitness-Funktion genannt) ein Wert bestimmt.
Selektion: Zufällige Auswahl von Lösungskandidaten aus der vorhandenen Lösungskandidatenmenge. Dabei werden Lösungskandidaten mit besseren Zielfunktionswerten mit einer höheren Wahrscheinlichkeit ausgewählt.

Rekombination: die Genome verschiedener Individuen werden gemischt und aus den neuen Parametern eine neue Generation von Individuen erzeugt (Vermehrung)
Mutation: Zufällige Veränderung der Wertekombinationen der Individuen der neuen Generation.
Nach einem bestimmten Verfahren wird die Menge der neuen Individuen aus der Menge der alten Individuen und der Menge der mutierten Nachfolger der Gewinner der Menge der alten Individuen gebildet. Der Algorithmus wird anschließend ab Schritt 2 wiederholt, oder nach einem Abbruchkriterium beendet und der beste verfügbare Lösungskandidat als Lösung definiert.

Rekursion:

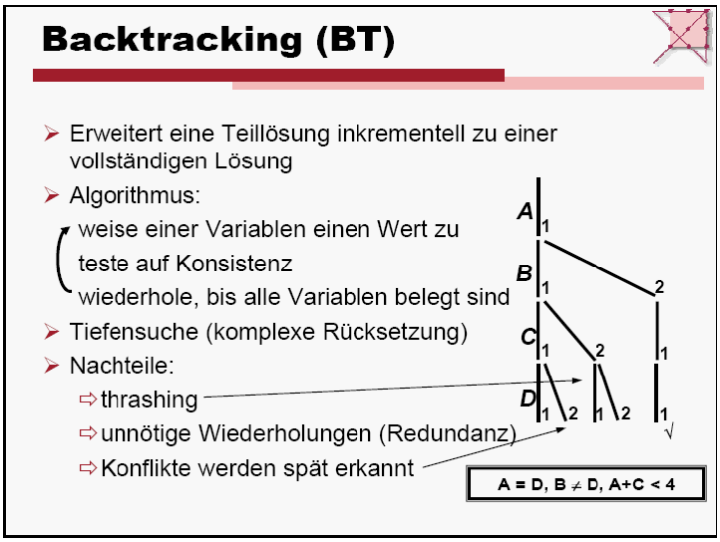
Rekursion, auch Rekurrenz oder Rekursivität (von lateinisch recurrere = zurücklaufen) ist ein allgemeines Prinzip zur Lösung von Problemen. In vielen Fällen ist die Rekursion eine von mehreren möglichen Problemlösungsstrategien, sie führt oft zu „eleganten“ mathematischen Lösungen. Als Rekursion bezeichnet man den Aufruf oder die Definition einer Funktion durch sich selbst. Ohne geeignete Abbruchbedingung geraten solche rückbezüglichen Aufrufe in einen so genannten infiniten Regress (umgangssprachlich Endlosschleife).

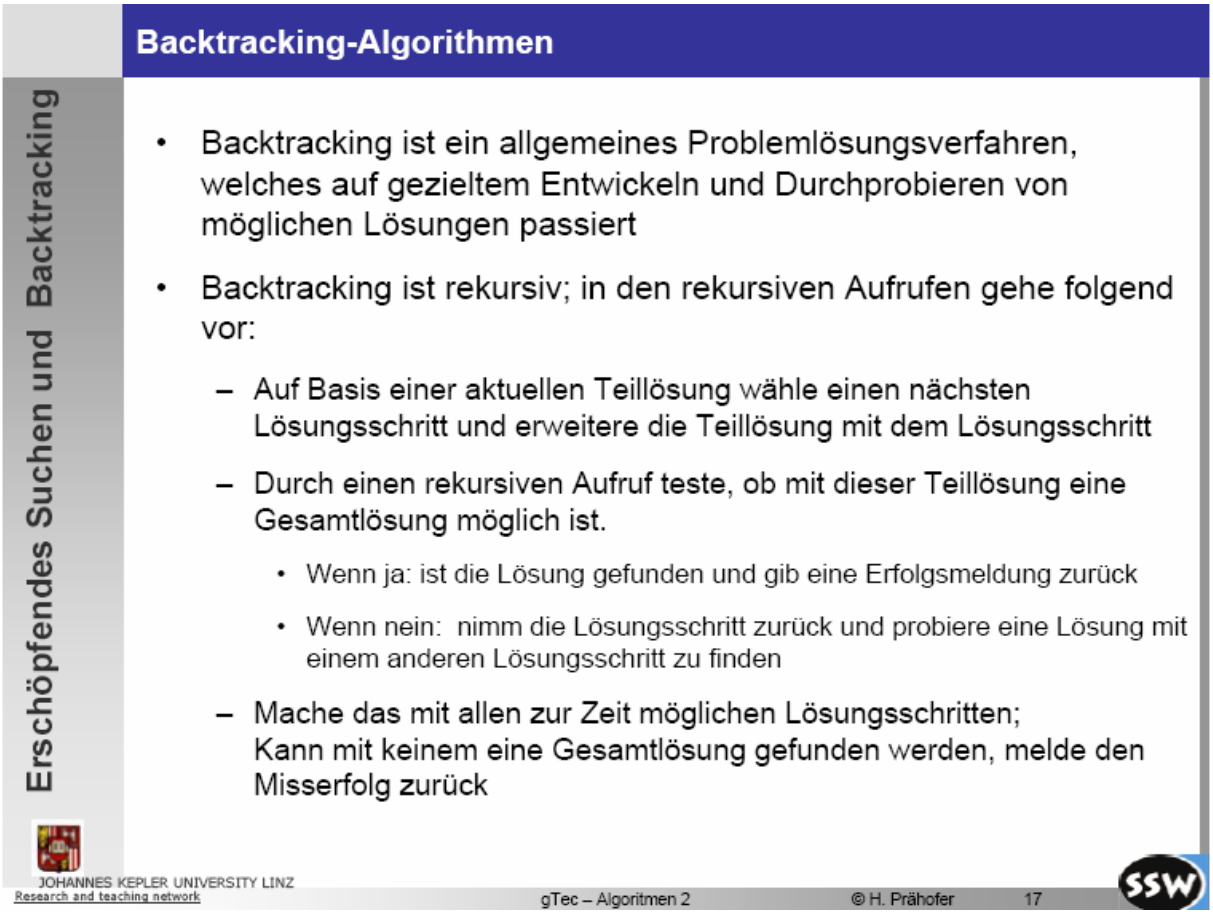
Andere Methoden:

Ein 'iterativer Reparaturalgorithmus' beginnt mit einer beliebigen Stellung der Damen auf dem Brett. Es zählt dann die Anzahl der Konflikte, und versucht, durch Umpositionieren der Damen die Anzahl der Konflikte zu reduzieren. Effizient ist etwa, die Dame mit den meisten Konflikten senkrecht auf die Position zu verschieben, auf der die geringste Anzahl von Konflikten auftritt. Mit dieser Methode kann das 1.000.000-Damenproblem, ausgehend von einer "vernünftigen" Versuchsposition gelöst werden. Derart große Bretter lassen sich mit expliziten Konstruktionsalgorithmen nicht lösen; allerdings kann der Iterationsalgorithmus nicht mit Sicherheit eine Lösung finden.

Für viele Probleme gibt es in der Informatik keine effizienten Algorithmen. Der natürlichste und einfachste Ansatz zur algorithmischen Lösung eines Problems besteht dann darin, einfach alle potenziellen Lösungen durchzuprobieren. Diese Methode nennt man „Brute Force“.

Backtracking:





Quellen:
http://de.wikipedia.org/wiki/Damenproblem
http://www.google.com/search?q=backtracking+n+dame&hl=de&lr=&domains=http://de.wikipedia.org&start=20&sa=N
Beispiel für generische Programmierung: http://www.dossier-andreas.net/ai/ga.html

2) Erkenntnis des n – Damen Problems

Da Backtacking am einfachsten zu realisieren ist und das n-Damenproblem auch maximal 12x12 Felder groß sein soll, ist der Lösungsweg dieses Verfahrens im Laufzeit - Entwicklungszeitverhältnis am effizientesten. Würde n größere Werte annehmen, wäre Backtracking sehr ineffizient.