Die Vor - und Nachteile der greedy strategy

„Erledige immer als nächstes den noch nicht bearbeiteten fettesten (d.h. größten oder kleinsten, teuersten, billigsten, optimalsten, ...) Teilbrocken des Problems.“

Erledigt man immer als nächstes den noch nicht bearbeiteten „*fettesten Teilbrocken*" des Problems, dann muss das nicht der für das Problem günstigste sein (siehe die einfache fuelle – Funktion). Daher führt die gierige Strategie in vielen Fällen zu recht guten, aber nicht unbedingt optimalen Lösungen. Warum kann diese „*gierige*" Auswahl zu negativen Effekten führen ?

Der Grund dafür ist für die Frage der Einordnung in die Suchverfahren von zentraler Bedeutung: Eine einmal bei einer Verzweigung gefällte Entscheidung wird später nicht wieder rückgängig gemacht. So können Entscheidungen, die **lokal günstig** erschienen, die aus Sicht einer optimalen Lösung (*die man ja nicht kennt*) aber tatsächlich ungünstig waren, später nicht wieder rückgängig gemacht werden. Es ist nicht einmal so, dass bei späteren Entscheidungen auf mögliche frühere Fehlentscheidungen hin untersucht wird! Gerade hierin, in diesem **bedenkenlosen Vorwärtsgehen**, liegt der besondere Vorteil von greedy - Verfahren.

Labyrinthe

Untersuchen wir dafür einmal den Beispielfall der Suche in **Labyrinthen**. Man müsste so etwas wie eine Zielheuristik haben, beispielsweise die Annahme, alle Wege mit möglichst langen Gängen seien die besten. Es zeigt sich nun sehr schnell, dass solche Heuristiken so willkürlich sind, dass ein Erfolg auf diesem Wege vernachlässigbar unwahrscheinlich ist. Die Suche des Ausgangs in einem Labyrinth ist also sicher ein Problem, dass sich für greedy nicht eignet.

Um beim Bild zu bleiben: Greedy kann ich in einem Labyrinth versuchen, bei dem es von jedem Ort zu jedem anderen viele verschiedene Wege gibt (also auch viele Zyklen) und ich beispielsweise nicht den Ausgang, sondern das Ungeheuer suche und sozusagen *„immer dem Geruch nachgehe"*. Dann kann es mir passieren, dass ich vielleicht an der einen oder anderen Stelle eine falsche Entscheidung treffe, diese aber ohne Folgen bleibt, da ich auf einem späteren kleinen Umweg immer noch das Ziel erreichen kann: Der Weg ist dann sicher nicht optimal, das Verfahren liefert aber immer noch ein akzeptables Ergebnis bei einem vermutlich sehr viel geringeren Aufwand, als bei einer systematischen Suche.
Aber: *In welcher Richtung stinkts ?*

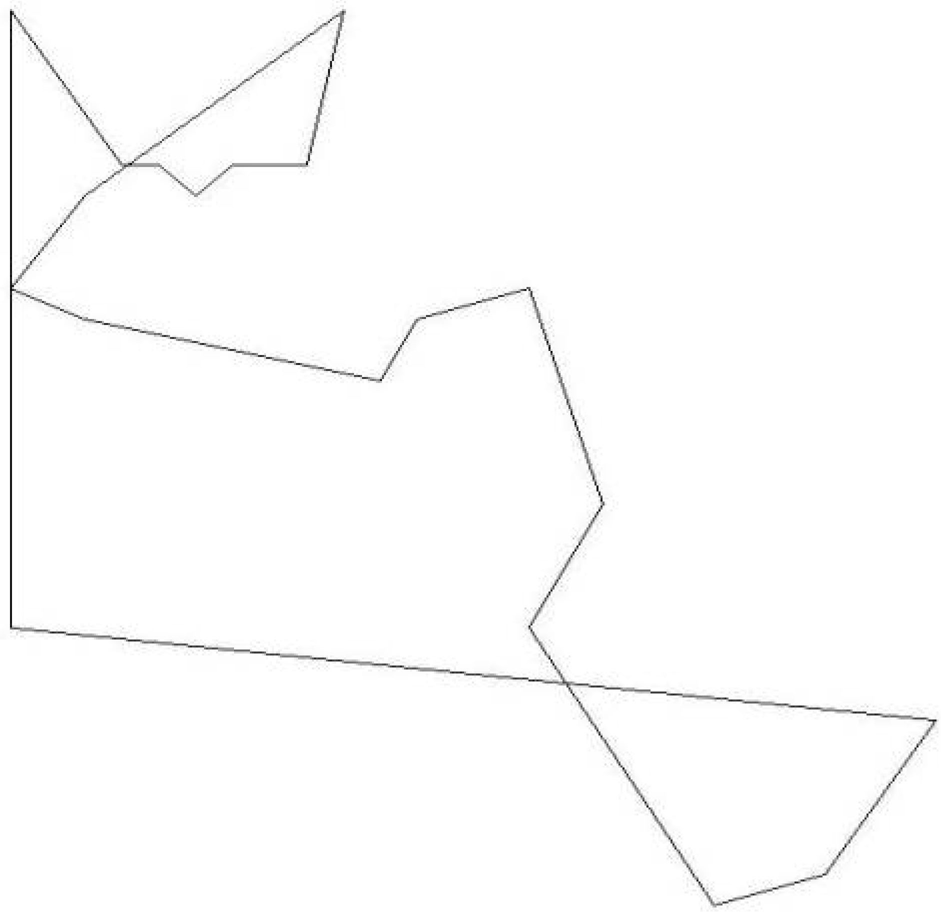
Bei der Entscheidung für eine Lösung mit greedy muss man vorher abklären, ob das Problem diesen Kriterien entspricht. Wenn ja, liegt man mit den Ergebnissen meistens recht gut. Das Problemn des **minimal spanning tree** ist dafür ein sehr gutes Beispiel.

Rundwege beim TSP

Beim **travelling salesperson problem** ist die Aufgabe zu lösen, dass die salesperson eine Rundreise minimaler Länge durchführt, die einerseits durch alle Städte führt, dabei soll aber jede Stadt genau einmal angefahren werden. Auf den ersten Blick ist das Problem nicht von den bisher betrachteten zu unterscheiden. Daher liegt es nahe, die Lösung nach einer dieser Strategien zu finden.

Wie auch immer man beginnt, mit irgend einem Knoten oder mit der kürzesten Kante im Graphen, wird man jeweils mit der kürzesten fortsetzenden Kante fortsetzen, die keine Zyklus bildet, es sei denn man hat bereits alle Knoten im Weg enthalten. Dann muss man die beiden „losen Enden“ verbinden, um den einen gewollten Zyklus zu erhalten.

Leider zeigt sich, dass dieser Algorithmus zwar sehr schnell und auch nicht so schlecht ist, aber nicht die optimale Lösung liefert. Ein Beispiel zeigt der folgende Graph, bei dem mögliche Verbesserungen „mit bloßem Auge“ zu erkennen sind.



Die Versuche, das Problem des tsp allgemein mit vollständigen (Problem: Größe des Suchraums) oder greedy Algorithmen zu lösen, führen in eine Sackgasse.

Bergsteige-Algorithmen (hill climbing)

Algorithmen, bei denen man Lösungen generiert, indem man stets den lokal günstigsten Nachfolger verfolgt, haben häufig ein Problem, das man gut mit dem Bild vom Bergsteigen beschreiben kann:
Wenn man stets das steilste Stück ansteigt, kann man eine Bergspitze erreichen, allerdings kann es dann sein, dass man in die Runde sieht und höhere Berge findet, zu denen man nur hätte gelangen können, wenn man zwischendurch auch einmal herabsteigt.
Man landet also in einem Nebenmaximum, nicht im Hauptmaximum des Suchraums.