Heuristic Optimisation

Part 1: Introduction

Sándor Zoltán Németh

http://web.mat.bham.ac.uk/S.Z.Nemeth

s.nemeth@bham.ac.uk

University of Birmingham

Overview

- Module structure
- Topics
- Assessment
- Difficult problems

Module structure

- Two lectures/week
- One tutorial/fortnight
- Office hours: Will be posted on CANVAS and my door: Room 324, Watson building
- Online material: CANVAS

Topics

1. Basic difficulties in problem solving. The need for heuristic optimization
2. Basic concepts: representation, optima, neighbourhood
3. Exhaustive search and local search
4. Greedy algorithm
5. Dynamic programming
6. A* search. Branch and bound algorithm
7. Simulated annealing
8. Tabu search
9. Evolutionary algorithms

Recommended books

Z. Michalewicz, D.B. Fogel: How to solve it: Modern heuristics, Springer, Corrected Third Printing 2002, ISBN 3-540-66061-5. (core text)

S. Russel, P. Norvig: Artificial Intelligence: A modern approach, Prentice Hall, Second edition, 2002, ISBN 0137903952. (10%)

Assessment

The details are on CANVAS.

What this course is and is not about

- It is about traditional and modern heuristic optimization methods for problem solving
- It is not about telling what the ultimate method to use is!

Example puzzle:

There is a well, open at the top, with diameter $D = 3m$. We throw two sticks of lengths $4m$ and $5m$ into the well. The sticks stay in a plane and cross each other. The problem is to find the distance h from the bottom of the well to the point where they cross.

Why are some problems difficult to solve?

- The size of the search space is huge
- The problem is very complicated, the solutions to a simplified model are useless
- The evaluation function varies with time, so a set of solutions is required
- There are heavy constraints, it is hard to even construct a feasible solution
- The person solving the problem is not prepared properly

The size of the search space

The Boolean satisfiability (SAT) problem

The task is to make a compound statement of Boolean variables evaluate to TRUE.

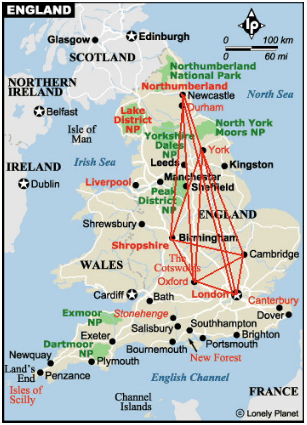
$F : \{0,1\}^n \rightarrow \{0,1\}$

For ex.
$F(x_1, x_2, \ldots, x_6) = (x_1 \bigvee \overline{x}_3 \bigvee x_5) \bigwedge (\overline{x}_2 \bigvee x_4 \bigvee x_6)$

For $n = 100$ the size of the search space is
$2^{100} \approx 10^{30}$!

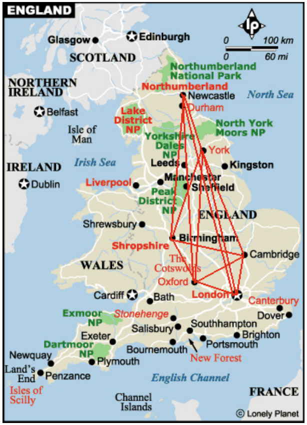
The size of the search space cont'd

The Traveling Salesperson (TSP) problem



The size of the search space cont'd

The Traveling Salesperson (TSP) problem



The traveling salesperson must visit every city in his territory exactly once and return to his hometown covering the shortest distance / shortest traveling time / using the least fuel.

The size of the search space cont'd

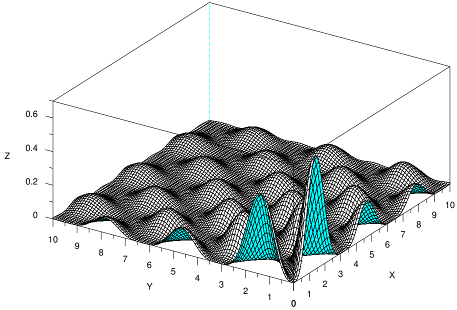
A nonlinear programming problem

Maximize function

$$G_2(x) = \left| \frac{\sum_{i=1}^{n} cos^4(x_i) - 2\prod_{i=1}^{n} cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$

subject to
$\prod_{i=1}^{n} x_i \geq 0.75$,
$\sum_{i=1}^{n} x_i \leq 7.5n$

with bounds $0 \leq x_i \leq 10$
for $1 \leq i \leq n$



Finding the model

$$\textit{Problem} \Rightarrow \textit{Model} \Rightarrow \textit{Solution}$$

Simplify the model and use a traditional optimizer:

$$\textit{Problem} \Rightarrow \textit{Model}_{approx} \Rightarrow \textit{Solution}_{prec}(\textit{Model}_{approx})$$

Keep the exact model and use a non-traditional optimizer to find a near-optimal solution:

$$\textit{Problem} \Rightarrow \textit{Model}_{prec} \Rightarrow \textit{Solution}_{approx}(\textit{Model}_{prec})$$

Change in time

TSP:

Possible routes might change due to rush hours

For ex. travel from A to B normally takes 40 minutes, but can be 60 minutes in worst case

Shall we use the expected time of 50 minutes in our calculations?

Constraints

The NLP problem:

We need the optimum within the feasible region

Hard constraints must be satisfied

Timetabling: a student cannot have two lectures at the same time

Soft constraints are desirable, but not mandatory

Timetabling: if there are 3 lectures/week for a module, they would be preferred on Monday, Wednesday and Friday

Heuristic optimization

The original Greek word means “I found it”

We use some information available about the problem

to reduce the region(s) of the search space
to be checked

or to speed up the search

at the cost of not guaranteeing the optimum

Summary

- Complex problems have a huge number of possible solutions
- We often need to simplify the problem, but then we might obtain useless solutions
- The problem's conditions can change in time.
- Real-world problems may have constraints, so generating feasible solutions is hard already!

Recommended reading

Z. Michalewicz & D.B. Fogel
How to Solve It: Modern Heuristics

Chapter 1. Why are some problems difficult to solve?