



Limits of Computation

17 - Common Problems Not Known to Be in P

Bernhard Reus

1



Last time

- how optimisation problems can be expressed as decision problems
- a number of famous, natural, “real-life” problems that are all provably in P



One more Problem in P (from last time)

3



Linear Programming

very versatile
& useful

- Solving linear inequalities
- **Instance:** a vector of *positive* (real) variables $0 \leq x$, row vector b , matrix A such that $Ax \leq b$ and column vector c
- **Question:** maximise $c^T x$

Simplex Algorithm

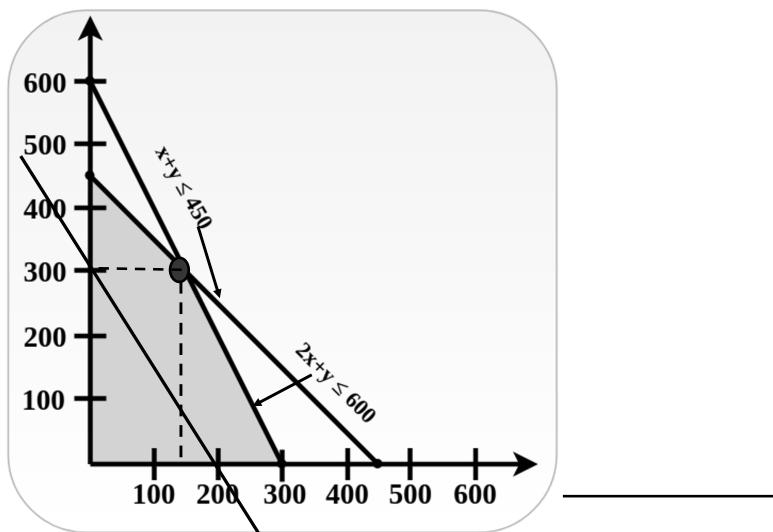
$$\begin{aligned} &\text{maximise} && P_1 \times x_1 + P_2 \times x_2 \\ &\text{where} && x_1 + x_2 \leq A \\ & && F_1 \times x_1 + F_2 \times x_2 \leq F \\ & && I_1 \times x_1 + I_2 \times x_2 \leq I \end{aligned}$$

$$\mathbf{b} = \begin{pmatrix} A \\ F \\ I \end{pmatrix} \quad A = \begin{pmatrix} 1 & 1 \\ F_1 & F_2 \\ I_1 & I_2 \end{pmatrix}$$
$$\mathbf{c} = (P_1 \ P_2)$$

Simplex algorithm does not have polynomial runtime (!) Karmarkar gave polynomial algorithm $O(n^{3.5} \times L^2 \times \ln L \times \ln \ln L)$ where n is the number of variables and L is size of input (binary).



Visualisation for Farmer



5



- Sorting
- Parsing
- Prime Number Test
- Graph Reachability
- Shortest Path
- Maximal Matching
- Max Flow-Min Cut
- Postman Problem
- Linear Programming

Problems in P

Attention: Most not presented as decision problems here, but class P is about decision problems only

and many, many more



Back to Today's Lecture

7

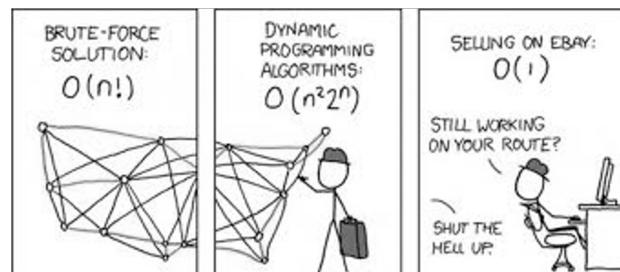


Complexity of natural problems

THIS TIME

e.g. finding the best route

- We introduce some more natural (and famous) problems and discuss their runtime complexity.
- For those problems today we don't know whether they are in **P** and the question remains: “*are they feasible?*”



<http://xkcd.com>

8



Problems that look intractable

- we introduce more problems, all highly relevant in practice.
- which can be solved easily by “brute-force” search algorithms but unfortunately these algorithms do not have polynomial runtime so they do **not** prove that the problems are in P.
- But it can be checked in P whether a potential solution is actually a solution (more about this next time).
- Now all problems (even optimisation problems) presented as decision problems for simplicity.

,



Problems of which we don't know whether they are in P

and that we conjecture
may be “intractable”



Travelling Salesman Problem

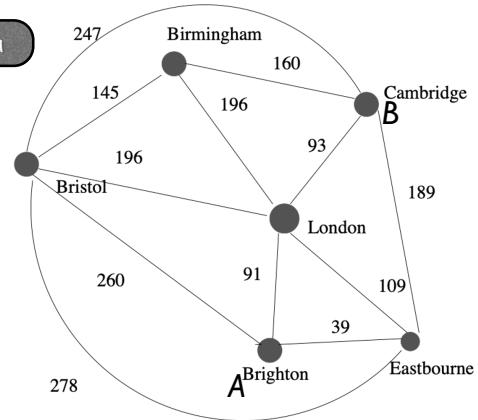
see Lecture 1, now presented as decision problem

- **Instance:**

a road map of cities, with distances attached to road segments, two cities A and B and a number K

- **Question:**

Is it possible to take a trip from A to B which passes through all cities and with a total trip length less or equal K (km) ?



distances in km in Lec. I we had $A=B$

in practice: for distribution of goods

II



Press coverage about TSP

guardian.co.uk

News | Sport | Comment | Culture | Business | Money | Life & style |

Environment > Bees

Bees' tiny brains beat computers, study finds

Bees can solve complex mathematical problems which keep computers busy for days, research has shown

24.10.2010



Researchers found that bees could solve the 'travelling salesman's' shortest route problem, despite having a brain the size of a grass seed. Photograph: Rex Features

Bees can solve complex mathematical problems which keep computers busy for days, research has shown.

The insects learn to fly the shortest route between flowers discovered in random order, effectively solving the "travelling salesman problem" said scientists at Royal Holloway, University of London.



How many colours?

- How many colours do you need to paint a map such that no adjacent countries have the same colour?
 - on the right 4 colours are used
 - are 3 colours enough?
 - 4 is *always* enough for maps; proved by K.Appel & W.Haken in 1977
["Every planar map is 4-colourable" Illinois J. Math. 21], already observed by Englishman Francis Guthrie in 1852
 - (fully machine-checked proof in 2004)



<http://www.doc.ic.ac.uk/~vf05/atp/casestudy.html>

maps are special graphs (planar graphs) where nodes are countries and edges between countries exist if the two countries have a common piece of borderline.

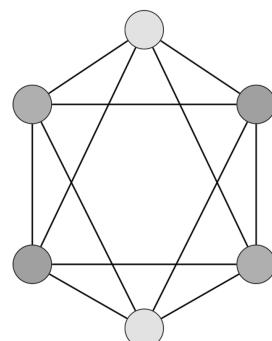
13



Graph Colouring Problem

- **Instance:**
a graph and a number K of colours to be used where K is greater or equal 3 (otherwise it's easy).

for maps the only interesting (hard) case is $K=3$



in practice: needed for register allocation in compilers, for traffic channel assignment (networks, air):
conflicting variables/planes/jobs have a common edge if they require the same resource (register/channel/machine);
 $\# \text{colours} = \text{how many resources needed at least at any given time?}$

- **Question:**
Can one colour the graph in a way that each pair of adjacent nodes (connected by an edge) uses two different colours (using only the K colours provided)?

14



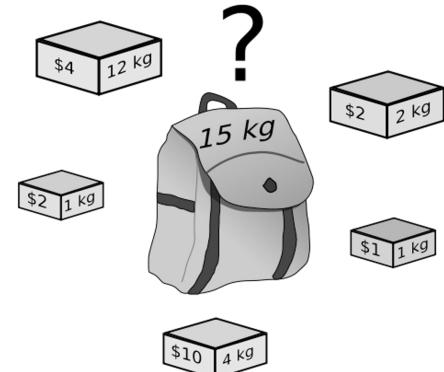
0-1 Knapsack Problem

- **Instance:**

a set of n items with sizes $\{w_1, \dots, w_n\}$ and profits $\{p_1, \dots, p_n\}$, the size of a knapsack W , and a threshold K . All values are positive natural numbers.

- **Question:**

Is there a way a subset of the n items can be packed such that their profits are at least K without exceeding the capacity W ?



javaingrab.blogspot.co.uk/2014/07/implementation-of-01-knapsack-problem.html

in practice: truck
loading, capacity planning

15

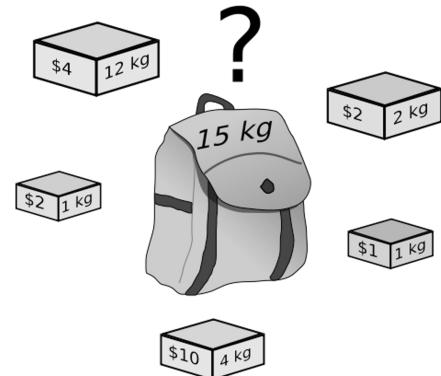


0-1 Knapsack Problem (cont'd)

Can be also represented via inequalities:

$$\sum_{i=1}^n w_i \times x_i \leq W \quad \sum_{i=1}^n p_i \times x_i \geq K$$

where the x_i prescribe the solution and must be all 0 or 1 (hence the name)



Proposition 17.1. Any instance of the 0-1 Knapsack Problem can be solved in $\mathcal{O}(n^2 \times p_{\max})$, where (as in Def. 17.4) n is the number of items and p_{\max} is the largest profit value.



Why is Knapsack not in P then?

16

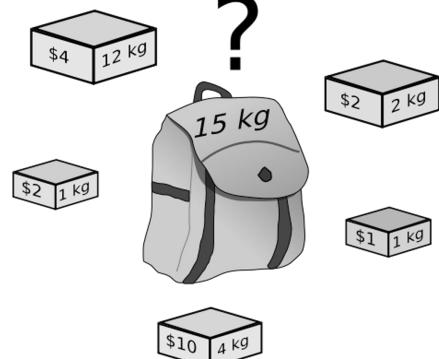


0-1 Knapsack Problem (cont'd)

Can be also represented via inequalities:

$$\sum_{i=1}^n w_i \times x_i \leq W \quad \sum_{i=1}^n p_i \times x_i \geq K$$

where the x_i prescribe the solution and must be all 0 or 1 (hence the name)



0-1 Knapsack is thus an instance of *Linear Programming* which we know is in **P**.

But all solutions **must** be integers, so we conclude that it is also unknown whether *Integer Linear Programming* is in **P**

17



Complexity of problems?

- We don't know whether the problems on previous slides are in **P**.
- Simple “generate and test” algorithms are exponential. For instance check all TSP paths: there are about $(n-1)!/2$ so time bound even worse than exponential 2^N .
- We define a new class for which we know that it contains all the problems just presented:
NEXT TIME!

n = size of graph
(number of nodes
i.e. cities)

Stirling's formula:
 $n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$

18



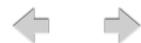
Problems not known to be in P

- TSP
- Graph Colouring
- 0-1 Knapsack Problem
- Linear Integer Programming

and many, many more

we'll meet some more later

19



END

© 2008-24. Bernhard Reus, University of Sussex

Next time:
Programs that can “guess”
and a *One Million Dollar*
Question!

20