Integer Programming Solvers

Jakub Mareček

School of Computer Science, The University of Nottingham
Jubilee Campus, Nottingham NG8 1BB, UK
E-mail: jxm@cs.nott.ac.uk

August 16, 2008

Motivation

- Combinatorial optimisation, combinatorial explosion
- Universal “solver”
- Solvers sacrificing optimality (local search)
- Solvers sacrificing performance (?!)
- Solvers and their “workhorses”

Linear Programming

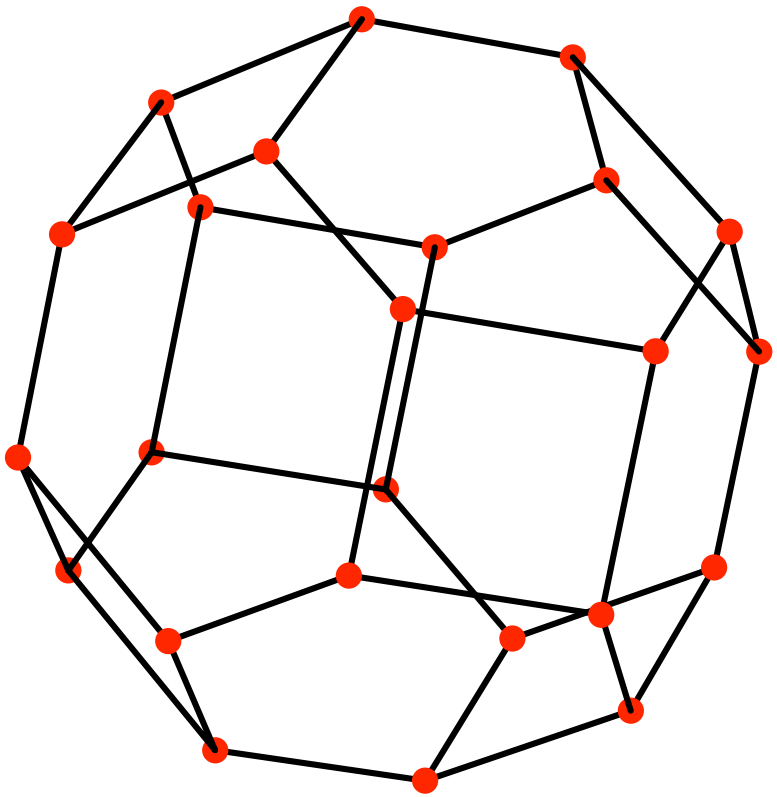
The Problem of Linear Programming

$$
\begin{array}{rrcl}
\text{objective}: \max & c_1x_1 + c_2x_2 + \cdots + & c_nx_n & \\
\text{constraints}: & a_{1,1}x_1 + a_{1,2}x_2 + \cdots + & a_{1,n}x_n \leq & b_1 \\
& a_{m,1}x_1 + a_{m,2}x_2 + \cdots + & a_{m,n}x_n \leq & b_m \\
& & x_1, \ldots, x_n \geq & 0
\end{array}
$$

- Modern solvers (CPLEX) run almost in linear time on average
- Structured instances with hundreds of millions of variables can be solved in hours

Linear Programming (Cntd.)

- Simplex: hopping over the corner points
- Interior point methods: go straight
- Ellipsoid methods: of theoretical interest



Integer Programming (IP)

The Problem of Integer Programming

$$
\begin{aligned}
\text{objective} : \max \quad & c_1 x_1 + c_2 x_2 + \cdots + c_n x_n \\
\text{constraints} : \quad & a_{1,1} x_1 + a_{1,2} x_2 + \cdots + a_{1,n} x_n \leq b_1 \\
& a_{m,1} x_1 + a_{m,2} x_2 + \cdots + a_{m,n} x_n \leq b_m \\
& x_1, \ldots, x_n \in \mathbb{Z}^+
\end{aligned}
$$

- Where is the difference?
- How important it is?

Integer Programming (Cntd.)

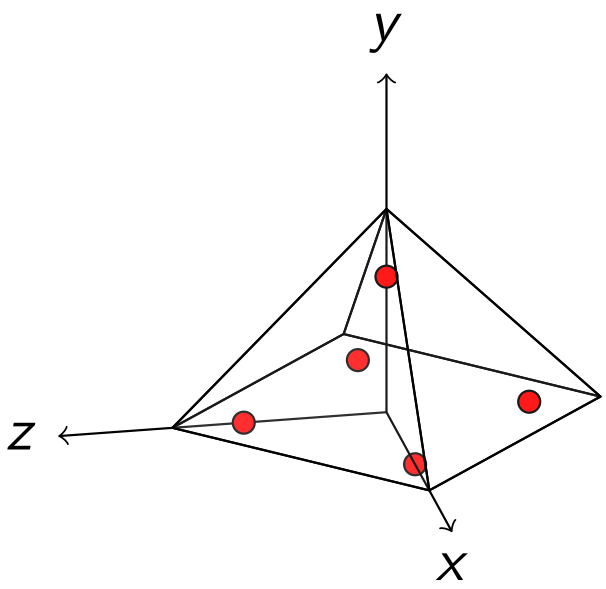


Figure: A poor linear programming relaxation.

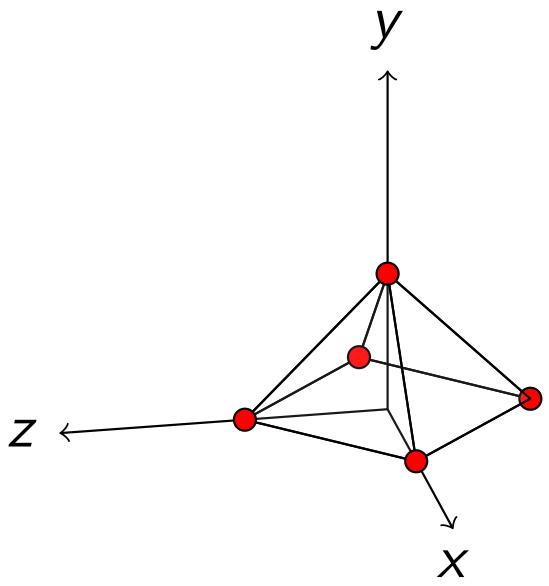


Figure: The convex hull of integer points. Typically, there are exponentially many facets.

Integer Programming (Cntd.)

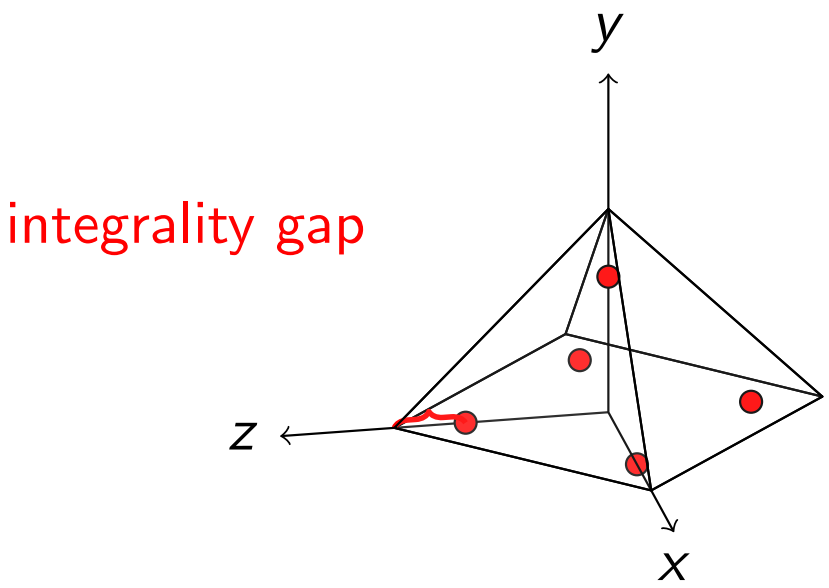


Figure: The "integrality gap" is the distance from the LP optimum to the IP optimum – or the value of the present best integer feasible solution.

Integer Programming (Cntd.)

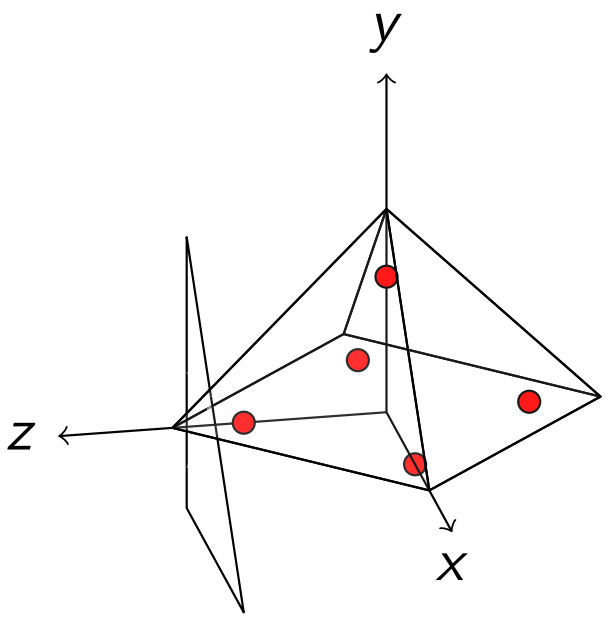


Figure: Pure cutting plane methods can go for ever ...

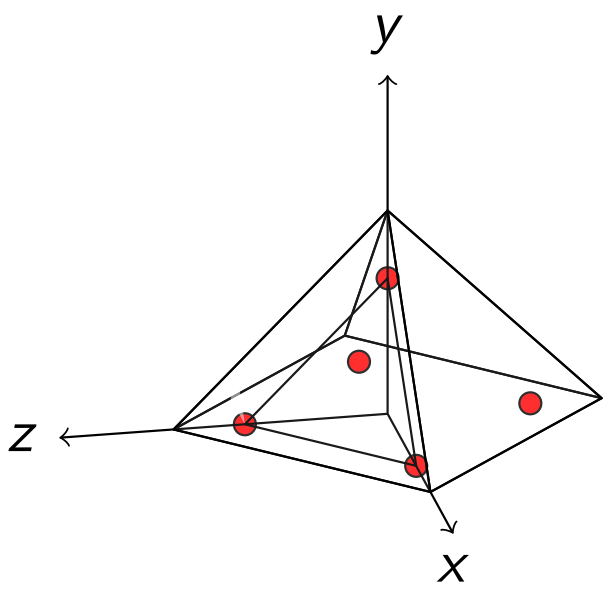


Figure: ... even if your cuts are facet-defining. Why?

Integer Programming (Cntd.)

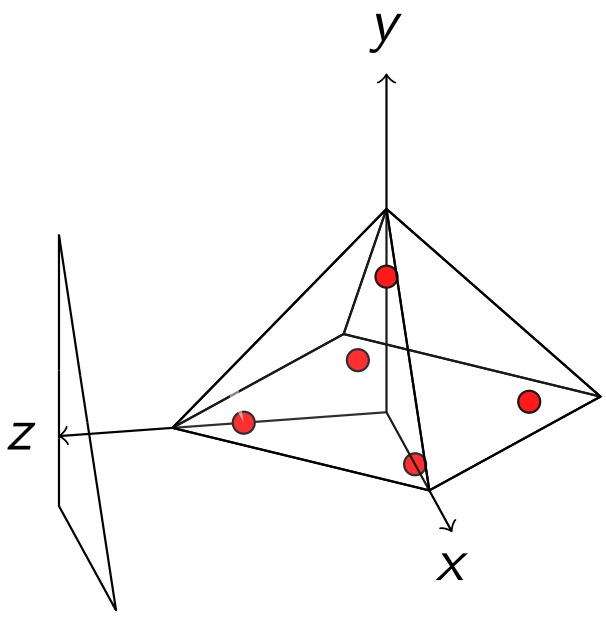


Figure: Branching can be thought of as a form of "divide and conquer". Here, branch on a variable by adding $x \geq 2$ ...

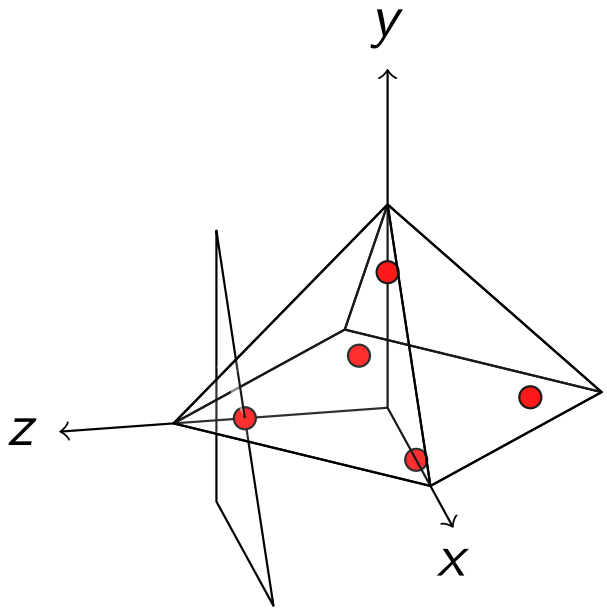


Figure: ... or by adding $x \leq 1$.

Integer Programming (Cntd.)

- Branch and bound: Building, pruning and traversing a tree
- Cuts by non-trivial rounding or polytope specific
- Loads of heuristics

- Branching heuristics: what (variable) to branch on, which direction to try first
- Primal heuristics: find the first feasible solution so that you can prune the tree
- Improvement heuristics: very often *the* source of feasible solutions
- Search strategies: which node in the tree should be visited next
- Cut generation: how much time to spend in separation, what cuts are good enough, when should they be removed, ...

What I am doing at Nottingham

Formerly:

- Preprocessing of a discrete set for polylog extreme ray queries
- Optimal transformation of graph colouring to multicolouring and applications

More recently:

- Primal heuristics for structured instances of binary integer programming
- Branching heuristics for binary integer programming
- Auto-tuning integer programming solvers

What I Was Doing Here

- An applet using VTK Java Wrappers
  Sends JVM crashing down
- Snippet for SVG and PDF generation from Jython
  Turns out to be rather bulky
- Suggestion: Sketch/TikZ output from plot3d
  Nils implemented a preliminary version
  Sketch turns out to be rather lame

`http://wiki.sagemath.org/JakubMarecek`

Conclusions

- For inapproximable problems, you need "heuristics"
- Design of "heuristics" often requires some insight
- Theoretical analysis is difficult, visualisations would help

See also:

- Sven Krumke: *Integer Programming*
  `optimierung.mathematik.uni-kl.de/~krumke/`
- Lex Schrijver: *A Course in Combinatorial Optimization*
  `www.cwi.nl/~lex/`
- Vašek Chvátal: *Linear Programming*
- Nemhauser & Wolsey: *Integer and Combinatorial Opt.*
- H. P. Williams: *Model Building in Mathematical Programming*

Many thanks!

- Many thanks to the organisers and you all!
- Any questions or comments are appreciated!

- My email addresses are at `http://cs.nott.ac.uk/~jxm`