60016 OPERATIONS RESEARCH

Integer Programming

16 November 2020

Second part of OR so far

- Duality
- Sensitivity and shadow prices
- Game theory (pure and mixed strategies)

Final topic: integer programming

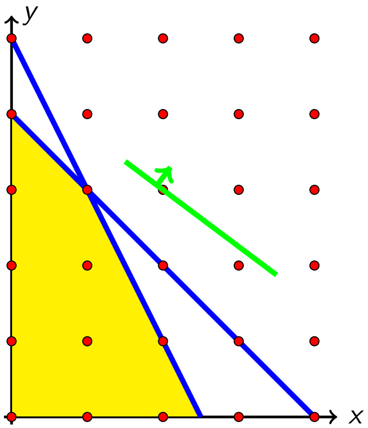
- **Integer Programming**
- **Integer Linear Programming (ILP)**
  - Mixed ILP
  - Pure ILP
  - ILP in the wild: `https://arxiv.org/abs/1706.07351`

Integer Programming

- Mathematical programming problems where one or more variables are integer valued:
  - Binary variables: $x_j \in \{0, 1\}$
    - e.g., take "yes or no" decisions
  - Integer variables: $x_j \in \{0, \ldots, n\}$
    - e.g., discrete amounts: cannot produce 3.6 cars
  - Programs can include both integer and real variables
    - e.g., Mixed Integer Linear Programming (MILP)

- Application areas:
  - Resource allocation
  - Traffic routing
  - Graph theory
  - Circuit design
  - ...

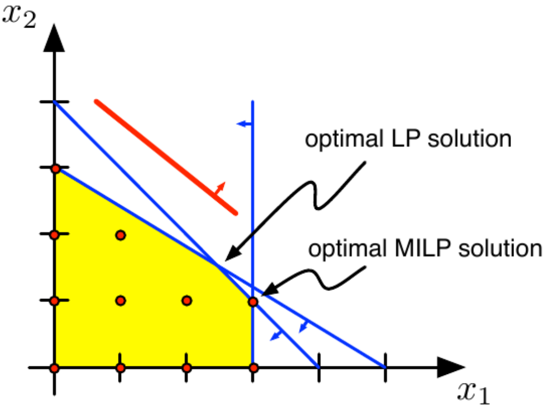
Warming up



$$\max_{x,y} 3x + 4y$$
$$\text{s.t.}\quad x + y \le 4$$
$$2x + y \le 5$$
$$x \ge 0,\ y \ge 0$$
$$\mathbf{x},\ \mathbf{y} \in \mathbb{N}_{\mathbf{0}}$$

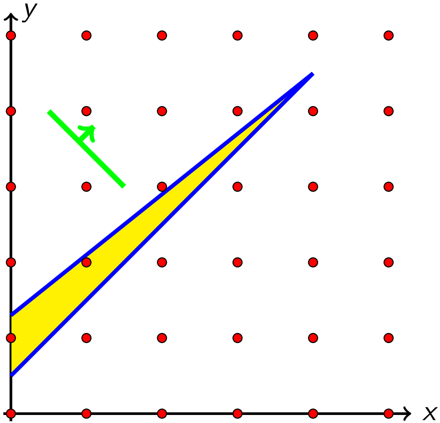
Sanity Check. Now what is the solution to the MILP? In general, will the solution of an MILP be the same as the LP constructed by dropping the integrality constraints?

Feasible Set with Integer Variables



Sanity Check. What is the relationship between minimum objective values at the MILP solution $f(x_{\mathrm{MILP}^*})$ the LP solution $f(x_{\mathrm{LP}^*})$ created by dropping integrality constraints?

Are MILP/LP solutions close? (example by H.P. Williams)



$$\begin{aligned} \max_{x,y} \quad & x_1 + \quad x_2 \\ \text{s.t.} \quad & -2x_1 + \ 2x_2 \geq 1 \\ & -8x_1 + 10x_2 \leq 13 \\ & x \geq 0,\ y \geq 0 \\ & x,\ y \in \mathbb{N}_0 \end{aligned}$$

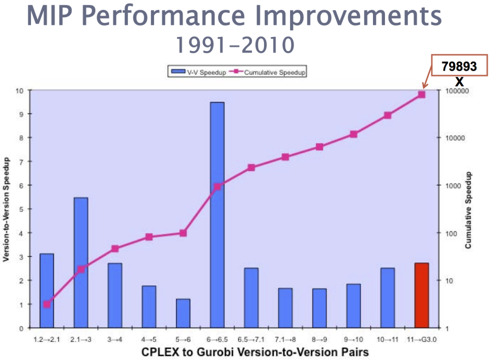
Sanity Check. Are the MILP and LP relaxed solutions in general close to one another?

Integer Programming

- **Common fallacy:** "Integer problems are easier to solve than continuous problems"
  - Integer problems can be very hard to solve!
  - n binary variables define 2^n possible combinations
- Integer linear programming is actively researched
  - Problems with 1000s of binary variables are solvable
  - Optimality gaps known for intermediate solutions
- Integer non-linear programming still a difficult area
  - Heuristics are often needed to aid solvers solve efficiently this classes of models.

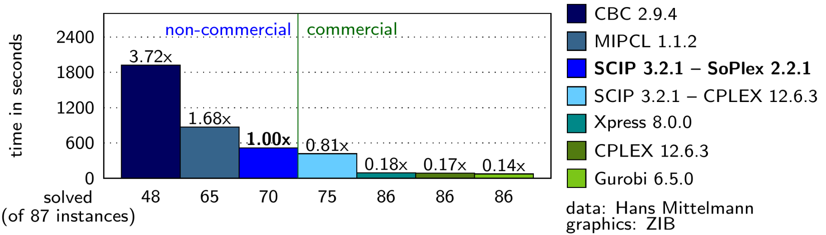
Figure: Vehicle Routing Problem: Major Need for Ocado, Tesco, etc.

State-of-the-art Solving Capability



- Graphic: Bob Bixby (CPLEX, Gurobi)
- 1852 Real-World MILPs;
- Parameter settings: Pure Defaults, 1 Thread, 3×10^4 s limit;
- All versions run on the same piece of hardware: CPLEX 1.2 (1991) – Gurobi 3.0 (2010).

State-of-the-art solvers combine many methods



Graphic: `http://scip.zib.de`

Nick Fury

- Early solvers mostly use branch & bound
- State-of-the-art solvers coordinate methods:
  - Branch & Bound;
  - Cutting planes, e.g. Gomory cuts;
  - Heuristics
- State-of-the-art solver Xpress written by developers in Birmingham, UK

Example 1: Capital Budgeting

- Company has resources $i \in \{1, \ldots, m\}$. Resource i has limited availability b_i.
- Company can undertake projects $j \in \{1, \ldots, n\}$. Project j requires a_{ij} units of resource i and gives revenues c_j.
- Which projects should be undertaken?

$$
\begin{aligned}
\max_{x} \quad & z = \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i && \forall\, i \in \{1, \ldots, m\} \\
& x_j \in \{0, 1\} && \forall\, j \in \{1, \ldots, n\}
\end{aligned}
$$

Example 2: Facility Location

- Company has m potential distribution sites $i \in \{1, \ldots, m\}$.
- Building a distribution centre at site i costs f_i.
- Company has n customers $j \in \{1, \ldots, n\}$ whose demands d_j need to be satisfied from one or more distribution centres.
- c_{ij}: cost to satisfy an amount x_{ij} of customer j's demand from distribution centre i, if centre i is built.
- Which distribution centres should be built, and how should the demand be satisfied, to minimise costs?

Example 2 (cont.): Facility Location

$$
\begin{aligned}
\min_{x,y} \quad & \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^{m} x_{ij} = d_j && \forall\, j \in \{1, \dots, n\} \\
& x_{ij} \leq d_j y_i && \forall\, i \in \{1, \dots, m\},\ j \in \{1, \dots, n\} \\
& x_{ij} \geq 0 && \forall\, i \in \{1, \dots, m\},\ j \in \{1, \dots, n\} \\
& y_i \in \{0, 1\} && \forall\, i \in \{1, \dots, m\}
\end{aligned}
$$

Example 3: Airline Crew Scheduling

- An airline wants to operate m flights per week
  - London-Madrid, Madrid-Paris, Paris-New York, ...
- Crews can be assigned to any of $j = 1, \dots, n$ flight sequences, each costing c_j
  - e.g., sequence {London-Madrid, Madrid-Paris}
  - $a_{ij} = 1$ if flight i is in sequence j, 0 otherwise
- $x_j = 1$ if a crew is assigned to flight sequence j, 0 otherwise
- Select what sequences to operate such that costs are minimal and the m flights all have a crew
- Decisal (`http://decisal.com`) is a London-based start-up solving planning, scheduling, and management problems for the airline industry

Example 3 (cont.): Airline Crew Scheduling

$$
\begin{aligned}
\min_{x} \quad & z = \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \geq 1 && \forall\, i \in \{1, \ldots, m\} \\
& x_j \in \{0, 1\} && \forall\, j \in \{1, \ldots, n\}
\end{aligned}
$$

Combinatorial Optimisation

- Combinatorial optimisation problems involve finding a optimal object from a finite set of objects.
  - A subarea of integer programming
  - Enumeration gets intractable as problem size grows.
- Problems often reducible to few categories:
  - Knapsack problem
  - Bin-Packing problem
  - Cutting stock problem
  - Minimum spanning tree problem
  - ...
- Special results and algorithms apply to these problems.

The Knapsack Problem

- Consider n items of weight w_j, $j \in \{1, \ldots, n\}$ and a knapsack of weight capacity W.
- Item j has value v_j, but not all items may fit the knapsack.
- How to maximise the total value of the knapsack?

$$\begin{aligned} \max_x \quad & z = \sum_{j=1}^{n} v_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} w_j x_j \leq W \\ & x_j \in \{0, 1\} \qquad\qquad \forall j \in \{1, \ldots, n\} \end{aligned}$$

The Bin-Packing Problem

- n items of weight w_j, $j \in \{1, \ldots, n\}$, k bins of capacity W
- $x_{ij} = 1$ if item j assigned to bin i, 0 otherwise
- Minimise the number of bins needed to store all items

$$
\begin{aligned}
\min_{x,y} \quad & z = \sum_{i=1}^{k} y_i \\
\text{s.t.} \quad & \sum_{j=1}^{n} w_j x_{ij} \leq W y_i \\
& \sum_{i=1}^{k} x_{ij} = 1 & \forall j \in \{1, \ldots, n\} \\
& x_{ij}, y_i \in \{0, 1\} & \forall i \in \{1, \ldots, k\}, \forall j \in \{1, \ldots, n\}
\end{aligned}
$$

General MILP Problems

- Mixed Integer Linear Programming (MILP) is the most general class of integer linear programming

$$
\begin{aligned}
\min\ z \quad &= c^{\mathrm{T}} x \\
\text{s.t.} \quad & Ax = b \\
& x_j \geq 0 && \text{for } j \in N = \{1, \ldots, n\} \\
& x_j \in \mathbb{N}_0 = \{0, 1, 2, \ldots\} && \text{for } j \in Z \subseteq N.
\end{aligned}
$$

  where $x_j \in N \setminus Z$ are continuous, as in LPs.
- Sometimes people call this class MIP (mixed-integer programming) rather than MILP.

Specialised Problems

MILP has several subareas of independent interest:

- Pure Integer Linear Programming (Pure ILP). $Z = N \cup \{z\}$, i.e., all variables (including slack and objective value) are integer.
- Binary Linear Programming (0-1 ILP). ILP where all variables are binary.
- Mixed Integer Binary Programming (MIBP): MILP where integer variables are binary, i.e., $x_j \in \{0, 1\}$ for $j \in Z$.

(Note: ILP often shortened to IP in daily use terminology, thus Pure IPs, 0-1 IPs, etc.)

MILP and Pure ILP Standard Forms

MILP standard form:

- Similar to LPs, in particular $b \geq 0$.
- Slack and excess variables in MILPs are continuous.

Pure IP standard form:

- Slack and excess variables in Pure IPs are integer-valued.
- Step 0. Apply LP standard form transformations, except addition of slack and excess variables, thus
  - Minimisation
  - Non-negative right-hand-sides
  - Free variables
- Step 1. Scale the equations of the model so that all coefficients are integers.
- Step 2. Insert integer slack and/or excess variables.

Example: Pure ILP Standard Form

Step 1. Scale the equations of the model so that all coefficients are integers:

$$\min \; z = -\frac{1}{3}x_1 - \frac{1}{2}x_2 \qquad (\times 6)$$

subject to

$$\begin{aligned} \frac{2}{3}x_1 + \frac{1}{3}x_2 &\leq \frac{4}{3} \qquad (\times 3) \\ \frac{1}{2}x_1 - \frac{3}{2}x_2 &\leq \frac{2}{3} \qquad (\times 6) \\ x_1, x_2 &\geq 0 \\ x_1, x_2 &\in \mathbb{N}_0. \end{aligned}$$

Example: Pure ILP Standard Form

Step 1. Scale the equations of the model so that all coefficients are integer:

$$\min \; z' = -2x_1 - 3x_2$$

subject to

$$\begin{aligned} 2x_1 + x_2 &\le 4 \\ 3x_1 - 9x_2 &\le 4 \\ x_1, x_2 &\ge 0 \\ x_1, x_2 &\in \mathbb{N}_0. \end{aligned}$$

where $z = z'/6$.

Example: ILP Standard Form

Step 2. Insert (integer) slack variables:

$$\min \; z' = -2x_1 - 3x_2$$

subject to

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 4 \\ 3x_1 - 9x_2 + x_4 &= 4 \\ x_1, x_2, x_3, x_4 &\geq 0 \\ x_1, x_2, x_3, x_4 &\in \mathbb{N}_0. \end{aligned}$$

Logical Operations: Either-Or

- We can model logical operations on the constraints via integer variables. For example, consider the expression

$$a_1^{\mathrm{T}} x \leq b_1 \quad \vee \quad a_2^{\mathrm{T}} x \leq b_2$$

- This can be expressed by:

$$\begin{aligned} a_1^{\mathrm{T}} x &\leq b_1 + M\delta \\ a_2^{\mathrm{T}} x &\leq b_2 + M(1-\delta) \\ \delta &\in \{0, 1\}\,, \end{aligned}$$

  where M is a large enough constant called "big-M".

- Sanity Check. If $\delta = 1$, which inequality is true?

Example: Either-Or

- We want to model the following problem:

$$\begin{aligned} &\min x \\ &\text{s.t. } x \in [0,1] \quad \vee \quad x \in [2,4]\,. \end{aligned}$$

- This can be expressed as:

$$\begin{aligned} \min\ & x \\ \text{s.t. } & x \geq 0 \\ & x \leq 4 \\ & x \leq 1 + M\delta \\ & x \geq 2 - M(1-\delta) \end{aligned}$$

- Sanity Check. How could we model exclusive or?

Logical Operations: "k-out-of-m"

- Satisfy at least k out of m constraints:

$$a_1^{\mathrm{T}} x \leq b_1,\ a_2^{\mathrm{T}} x \leq b_2,\ \ldots,\ a_m^{\mathrm{T}} x \leq b_m$$

- This can be expressed by:

$$\begin{aligned} a_1^{\mathrm{T}} x &\leq b_1 + M\delta_1 \\ &\vdots \\ a_m^{\mathrm{T}} x &\leq b_m + M\delta_m \\ \sum_{j=1}^{m} \delta_j &\leq m - k \\ \delta_j &\in \{0, 1\}, \forall j \in \{1, \ldots, m\} \end{aligned}$$

Finite-Valued Variables

- Assume a variable x_j can only take a finite number of values: $x_j \in \{p_1, \dots, p_m\}$.
- We can introduce variables $z_{j1}, \dots, z_{jm} \in \{0, 1\}$ and add the constraint

$$z_{j1} + \dots + z_{jm} = 1 \qquad (*)$$

- Now, we can replace

$$x_j = p_1 z_{j1} + \dots + p_m z_{jm}$$

  in the objective function and all constraints.
- Due to $(*)$, x_j can only assume a single value

Example: Finite-Valued Variables

Consider the following problem:

$$\min \; z = -2x_1 - 3x_2$$

subject to

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 4 \\ 3x_1 - 9x_2 + x_4 &= 4 \\ x_1, x_2, x_3, x_4 &\geq 0 \\ x_1 &\in \{1, 3, 11\} \\ x_2, x_3, x_4 &\in \mathbb{N}_0. \end{aligned}$$

Example: Finite-Valued Variables

Replace $x_1 = z_{11} + 3z_{12} + 11z_{13}$ everywhere:

$$\min\ z = -2z_{11} - 6z_{12} - 22z_{13} - 3x_2$$

subject to

$$\begin{aligned}
2z_{11} + 6z_{12} + 22z_{13} + x_2 + x_3 &= 4 \\
3z_{11} + 9z_{12} + 33z_{13} - 9x_2 + x_4 &= 4 \\
z_{11} + z_{12} + z_{13} &= 1 \\
x_2, x_3, x_4 &\geq 0 \\
z_{11}, z_{12}, z_{13} &\in \{0, 1\} \\
x_2, x_3, x_4 &\in \mathbb{N}_0.
\end{aligned}$$