Introduction to integer linear programming

Martin Branda

Charles University in Prague
Faculty of Mathematics and Physics
Department of Probability and Mathematical Statistics

Computational Aspects of Optimization

Knapsack problem

Values $a_1 = 4$, $a_2 = 6$, $a_3 = 9$, costs $c_1 = 4$, $c_2 = 6$, $c_3 = 11$, budget $b = 10$:

$$\begin{aligned} \max \ & \sum_{i=1}^{3} c_i x_i \\ \text{s.t. } & \sum_{i=1}^{3} a_i x_i \leq 10, \\ & x_i \in \{0, 1\}. \end{aligned}$$

Consider $=$ instead of $\leq$, or $0 \leq x_i \leq 1$ instead of $x_i \in \{0, 1\}$ ...

Why is integrality so important?

Real (mixed-)integer programming problems (not always linear)

- **Portfolio optimization** – integer number of assets, fixed transaction costs
- **Scheduling** – integer (binary) decision variables to assign a job to a machine
- **Vehicle Routing Problems** (VRP) – binary decision variables which identify a successor of a node on the route
- ...

In general – modelling of **logical relations**, e.g.

- at least two constraints from three are fulfilled,
- if we buy this asset than the fixed transaction costs increase,
- ...

Integer linear programming

$$\min c^T x \tag{1}$$
$$Ax \geq b, \tag{2}$$
$$x \in \mathbb{Z}^n_+. \tag{3}$$

Assumption: all coefficients are integer (rational before multiplying by a proper constant).

Set of feasible solution and its relaxation

$$S = \{x \in \mathbb{Z}^n_+ : \; Ax \geq b\}, \tag{4}$$
$$P = \{x \in \mathbb{R}^n_+ : \; Ax \geq b\} \tag{5}$$

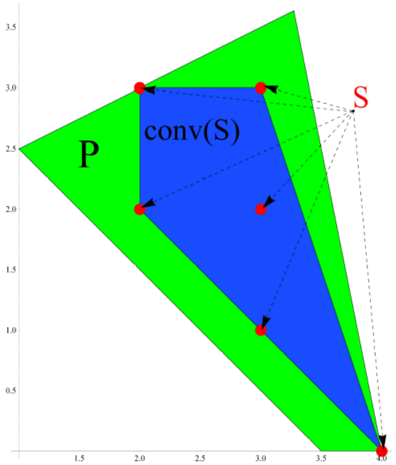
Obviously $S \subseteq P$. Not so trivial that $S \subseteq \mathrm{conv}(S) \subseteq P$.

Example

Consider set S given by

$$\begin{aligned} x_1 - 2x_2 &\geq -4, && (6)\\ -5x_1 - x_2 &\geq -20, && (7)\\ 2x_1 + 2x_2 &\geq 7, && (8)\\ x_1, x_2 &\in \mathbb{Z}_+. && (9) \end{aligned}$$

Set of feasible solutions, its relaxation and convex envelope



Škoda (2010)

Integer linear programming problem

Problem

$$\min c^T x : \; x \in S. \tag{10}$$

is equivalent to

$$\min c^T x : \; x \in \mathrm{conv}(S). \tag{11}$$

$\mathrm{conv}(S)$ is very difficult to construct – many constraints ("strong cuts") are necessary (there are some exceptions).

LP-relaxation:

$$\min c^T x : \; x \in P. \tag{12}$$

Mixed-integer linear programming

Often both integer and continuous decision variable appear:

$$\begin{aligned} \min\ & c^T x + d^T y \\ \text{s.t. } & Ax + By \geq b \\ & x \in \mathbb{Z}^n_+,\ y \in \mathbb{R}^{n'}_+. \end{aligned}$$

(WE DO NOT CONSIDER IN INTRODUCTION)

Basic algorithms

We consider:

- **Cutting Plane Method**
- **Branch-and-Bound**

There are methods combining previous alg., e.g. Branch-and-Cut.

Cutting plane method – Gomory cuts

1. Solve LP-relaxation using (primal or dual) SIMPLEX algorithm.
   - If the solution is integral – END, we have found an optimal solution,
   - otherwise continue with the next step.
2. Add a **Gomory cut** (...) and solve the resulting problem using DUAL SIMPLEX alg.

Example

$$\min 4x_1 + 5x_2 \tag{13}$$
$$x_1 + 4x_2 \geq 5, \tag{14}$$
$$3x_1 + 2x_2 \geq 7, \tag{15}$$
$$x_1, x_2 \in \mathbb{Z}^n_+. \tag{16}$$

Dual simplex for LP-relaxation ...

After two iterations of the dual SIMPLEX algorithm ...

			4	5	0	0
			x_1	x_2	x_3	x_4
5	x_2	8/10	0	1	-3/10	1/10
4	x_1	18/10	1	0	2/10	-4/10
		112/10	0	0	-7/10	-11/10

Gomory cuts

There is a row in simplex table, which corresponds to a non-integral solution x_i in the form:

$$x_i + \sum_{j \in N} w_{ij} x_j = d_i, \tag{17}$$

where N denotes the set of non-basic variables; d_i is non-integral. We denote

$$w_{ij} = \lfloor w_{ij} \rfloor + f_{ij}, \tag{18}$$

$$d_i = \lfloor d_i \rfloor + f_i, \tag{19}$$

i.e. $0 \leq f_{ij}, f_i < 1$.

$$\sum_{j \in N} f_{ij} x_j \geq f_i, \tag{20}$$

or rather $-\sum_{j \in N} f_{ij} x_j + s = -f_i$, $s \geq 0$.

Gomory cuts

General properties of cuts (including Gomory ones):

- Property 1: Current (non-integral) solution becomes infeasible (it is cut).
- Property 2: No feasible integral solution becomes infeasible (it is not cut).

Gomory cuts – property 1

We express the constraints in the form

$$x_i + \sum_{j\in N} (\lfloor w_{ij} \rfloor + f_{ij}) x_j \quad = \quad \lfloor d_i \rfloor + f_i, \tag{21}$$

$$x_i + \sum_{j\in N} \lfloor w_{ij} \rfloor x_j - \lfloor d_i \rfloor \quad = \quad f_i - \sum_{j\in N} f_{ij} x_j. \tag{22}$$

Current solution $x_j^* = 0$ pro $j \in N$ a $x_i^* = d_i$ is non-integral, i.e. $0 < x_i^* - \lfloor d_i \rfloor < 1$, thus

$$0 < x_i^* - \lfloor d_i \rfloor = f_i - \sum_{j\in N} f_{ij} x_j^* \tag{23}$$

and

$$\sum_{j\in N} f_{ij} x_j^* < f_i, \tag{24}$$

which is a contradiction with Gomory cut.

Gomory cuts – property 2

Consider an arbitrary integral feasible solution and rewrite the constraint as

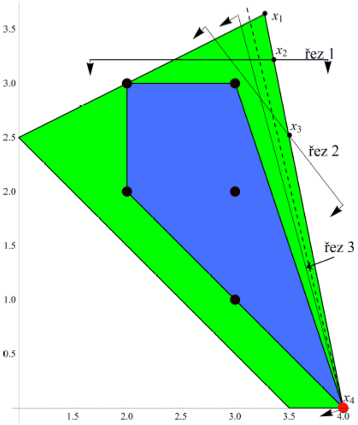
$$x_i + \sum_{j \in N} \lfloor w_{ij} \rfloor x_j - \lfloor d_i \rfloor \;=\; f_i - \sum_{j \in N} f_{ij} x_j, \tag{25}$$

Left-hand side (LS) is integral, thus right-hand side (RS) is integral. Moreover, $f_i < 1$ a $\sum_{j \in N} f_{ij} x_j \geq 0$, thus RS is strictly lower than 1 and at the same time it is integral, thus lower or equal to 0, i.e. we obtain Gomory cut

$$f_i - \sum_{j \in N} f_{ij} x_j \leq 0. \tag{26}$$

Thus each integral solution fulfills it.

Cutting plane methods – steps



Škoda (2010)

Dantzig cuts

$$\sum_{j \in N} x_j \geq 1. \qquad (27)$$

(Remind that non-basic variables are equal to zero.)

After two iterations of the dual SIMPLEX algorithm ...

			4	5	0	0
			x_1	x_2	x_3	x_4
5	x_2	8/10	0	1	-3/10	1/10
4	x_1	18/10	1	0	2/10	-4/10
		112/10	0	0	-7/10	-11/10

For example, x_1 is not integral:

$$\begin{aligned} x_1 + 2/10x_3 - 4/10x_4 &= 18/10, \\ x_1 + (0 + 2/10)x_3 + (-1 + 6/10)x_4 &= 1 + 8/10. \end{aligned}$$

Gomory cut:

$$2/10x_3 + 6/10x_4 \geq 8/10.$$

New simplex table

			4	5	0	0	0
			x_1	x_2	x_3	x_4	x_5
5	x_2	8/10	0	1	-3/10	1/10	0
4	x_1	18/10	1	0	2/10	-4/10	0
0	x_5	-8/10	0	0	- 2/10	-6/10	1
		112/10	0	0	-7/10	-11/10	0

Dual simplex alg. ...

Branch-and-Bound

General principles:

- Solve LP problem without integrality only.
- Branch using additional constraints on integrality: $x_i \leq \lfloor x_i^* \rfloor$, $x_i \geq \lfloor x_i^* \rfloor + 1$.
- Cut inperspective branches before solving (using bounds on the optimal value).

Branch-and-Bound

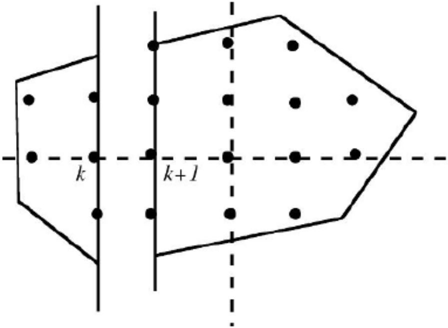
General principles:

- Solve only LP problems with relaxed integrality.
- **Branching**: if an optimal solution is not integral, e.g. $\hat{x}_i$, create and save two new problems with constraints $x_i \leq \lfloor \hat{x}_i \rfloor$, $x_i \geq \lceil \hat{x}_i \rceil$.
- **Bounding** ("different" cutting): save the objective value of the best integral solution and cut all problems in the queue created from the problems with higher optimal values[1].

Exact algorithm . .

[1] Branching cannot improve it.

Branch-and-Bound



P. Pedegral (2004). Introduction to optimization, Springer-Verlag, New York.

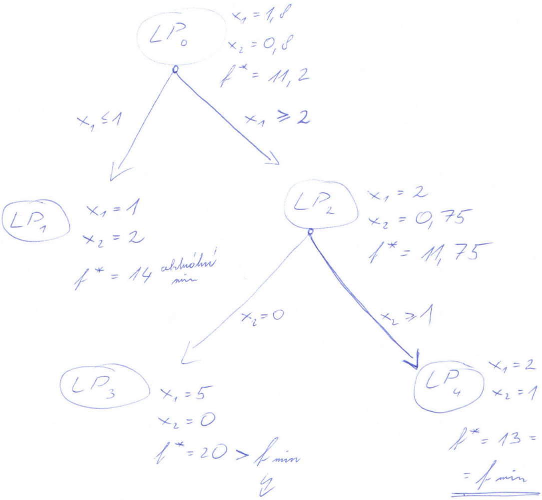
Branch-and-Bound

0. $f_{min} = \infty$, $x_{min} = \cdot$, list of problems $P = \emptyset$
   Solve LP-relaxed problem and obtain f^*, x^*. If the solution is integral, STOP. If the problem is infeasible or unbounded, STOP.
1. **BRANCHING**: There is x_i basic non-integral variable such that $k < x_i < k+1$ for some $k \in \mathbb{N}$:
   - Add constraint $x_i \leq k$ to previous problem and put it into list P.
   - Add constraint $x_i \geq k+1$ to previous problem and put it into list P.
2. Take problem from P and solve it: f^*, x^*.
3.
   - If $f^* < f_{min}$ and x^* is non-integral, GO TO 1.
   - **BOUNDING**: If $f^* < f_{min}$ a x^* is integral, set $f_{min} = f^*$ a $x_{min} = x^*$, GO TO 4.
   - **BOUNDING**: If $f^* \geq f_{min}$, GO TO 4.
   - Problem is infeasible, GO TO 4.
4.
   - If $P \neq \emptyset$, GO TO 2.
   - If $P = \emptyset$ a $f_{min} = \infty$, integral solution does not exist.
   - If $P = \emptyset$ a $f_{min} < \infty$, optimal value and solution are f_{min}, x_{min}.

Better ...

2./3. Take problem from list P and solve it: f^*, x^*. If for the optimal value of the current problem holds $f^* \geq f_{min}$, then the branching is not necessary, since by solving the problems with added branching constraints we can only increase the optimal value and obtain the same f_{min}.

Branch-and-Bound



Branch-and-Bound

Algorithmic issues:

- **Problem selection from list** P: FIFO/LIFO/problem with the smallest f^*.
- **Selection of the branching variable** x_i^*: the highest/smallest violation of integrality OR the highest/smallest coefficient in the objective function.

Totally unimodular matrix

Totally unimodular matrix A: for arbitrary INTEGRAL right-hand side vector b we obtain an integral solution, e.g. transportation problem.

Algorithms – a remark

(Relative) difference between a lower and upper bound – construct the upper bound (for minimization) using a feasible solution, lower bound ?

Duality

Set $S(b) = \{x \in \mathbb{Z}^n_+ : \; Ax = b\}$ and define the **value function**

$$z(b) = \min_{x \in S(b)} c^T x. \tag{28}$$

A dual function $F : \mathbb{R}^m \to \overline{\mathbb{R}}$

$$F(b) \leq z(b), \; \forall b \in \mathbb{R}^m. \tag{29}$$

A general form of **dual problem**

$$\max_F \{F(b) : \; \text{s.t. } F(b) \leq z(b), \; b \in \mathbb{R}^m, \; F : \mathbb{R}^m \to \mathbb{R}\}. \tag{30}$$

We call F a **weak dual** function if it is feasible, and **strong dual** if moreover $F(b) = z(b)$.

Duality

A function F is **subadditive** over a domain Θ if

$$F(\theta_1 + \theta_2) \le F(\theta_1) + F(\theta_2)$$

for all $\theta_1 + \theta_2, \theta_1, \theta_2 \in \Theta$.

The value function z is subadditive over $\{b : S(b) \neq \emptyset\}$, since the sum of optimal x's is feasible for the problem with $b_1 + b_2$ r.h.s., i.e. $\hat{x}_1 + \hat{x}_2 \in S(b_1 + b_2)$.

Duality

If F is subadditive, then condition $F(Ax) \leq c^T x$ for $x \in \mathbb{Z}^n_+$ is equivalent to $F(a_{.j}) \leq c_j$, $j = 1, \ldots, m$.

This is true since $F(Ae_j) \leq c^T e_j$ is the same as $F(a_{.j}) \leq c_j$.

On the other hand, if F is subadditive and $F(a_{.j}) \leq c_j$, $j = 1, \ldots, m$ imply

$$F(Ax) \leq \sum_{j=1}^{m} F(a_{.j}) x_j \leq \sum_{j=1}^{m} c_j x_j = c^T x.$$

Duality

If we set

$$\Gamma^m = \{F : \mathbb{R}^m \to \mathbb{R},\ F(0) = 0,\ F \text{ subadditive}\},$$

then we can write a **subadditive dual** independent of x:

$$\max_F \{F(b) :\ \text{s.t. } F(a_{.j}) \le c_j,\ F \in \Gamma^m\}. \tag{31}$$

Weak and strong duality holds.

An easy feasible solution based on LP duality (= weak dual)

$$F_{LP}(b) = \max_y b^T y \text{ s.t. } A^T y \le c. \tag{32}$$

Duality

Complementary slackness condition: if $\hat{x}$ is an optimal solution for IP, and $\hat{F}$ is an optimal subadditive dual solution, then

$$(\hat{F}(a_{\dots j}) - c_j)\hat{x}_j = 0,\ j = 1, \dots, m.$$

Software

... even for nonlinear integer problems

- **Interfaces**: GAMS, CPlex Studio, Gurobi, ...
- **Solvers**: CPlex (MILP, MIQP), Gurobi (MILP, MIQP), Baron, Bonmin (MINLP), Dicopt (MINLP), Knitro (MINLP), Lindo, ...

For difficult problems usually **heuristic and meta-heuristic algorithms** (greedy h., genetic alg., tabu search, simulated annealing, ...)

GAMS

Integer variables

- **Integer variables** – nonnegative with predefined upper bound 100 (can be changed using *x.up(i) = 1000;*)!
- **Binary variables**

Command SOLVE using

- MILP
- MIQCP
- MINLP

GAMS – options

TOLERANCE for optimal value of the integer problems:

- **optcr** – relative tolerance (default value 0.1 – usually too high)
- **optca** – absolute tolerance (turned off)
- **reslim** – maximal running time in seconds (default value 1000 – usually too low)
- **nlp** = *conopt* , **lp** = *gurobi* , **mip** = *cplex* – solver selection in code

For example
OPTIONS optcr=0.000001 reslim = 3600;

Literature

- G.L. Nemhauser, L.A. Wolsey (1989). Integer Programming. Chapter VI in Handbooks in OR & MS, Vol. 1, G.L. Nemhauser et al. Eds.
- P. Pedegral (2004). Introduction to optimization, Springer-Verlag, New York.
- Š. Škoda: Řešení lineárních úloh s celočíselnými omezeními v GAMSu. Bc. práce MFF UK, 2010. (In Czech)
- L.A. Wolsey (1998). Integer Programming. Wiley, New York.
- L.A. Wolsey, G.L. Nemhauser (1999). Integer and Combinatorial Optimization. Wiley, New York.