Tuesday, November 19

Program of the day:

- Branch-and-bound
- Using dual simplex to find bounds
- Hierarchy of techniques
- Preprocessing
- Example: A location problem solved through branch-and-bound

Duality

Branch-and-bound, economics: upper bound on LP.

$$\begin{array}{lrcrcrcrcr}
\text{maximize} & 4x_1 & + & x_2 & + & 5x_3 & + & 3x_4 & & \\
\text{subject to} & x_1 & - & x_2 & - & x_3 & + & 3x_4 & \leq & 1 \\
& 5x_1 & + & x_2 & + & 3x_3 & + & 8x_4 & \leq & 55 \\
& -x_1 & + & 2x_2 & + & 3x_3 & - & 5x_4 & \leq & 3 \\
& & & & & & x_1, x_2, & x_3, x_4 & \geq & 0
\end{array} \qquad (1)$$

Multiplying the second constraint by two

$$10x_1 + 2x_2 + 6x_3 + 16x_4 \leq 110$$

thus $z^* \leq 110$.
Linear combination of some constraints: second and third constraint

$$4x_1 + 3x_2 + 6x_3 + 3x_4 \leq 58$$

thus $z^* \leq 58$.
In general *any* linear combination.
multipliers y_1, y_2, y_3, demand $y_1, y_2, y_3 \geq 0$

$$\begin{aligned}
& y_1(x_1 - x_2 - x_3 + 3x_4) + \\
& y_2(5x_1 + x_2 + 3x_3 + 8x_4) + \\
& y_3(-x_1 + 2x_2 + 3x_3 - 5x_4) \leq y_1 + 55y_2 + 3y_3
\end{aligned}$$

which is equivalent to

$$\begin{array}{l}(y_1 + 5y_2 - y_3)x_1 + \\ (-y_1 + y_2 + 2y_3)x_2 + \\ (-y_1 + 3y_2 + 3y_3)x_3 + \\ (3y_1 + 8y_2 + 5y_3)x_4 \leq y_1 + 55y_2 + 3y_3\end{array} \tag{2}$$

coefficients must exceed those in (1):

$$\begin{array}{rcrcrcl} y_1 & + & 5y_2 & - & y_3 & \geq & 4 \\ -y_1 & + & y_2 & + & 2y_3 & \geq & 1 \\ -y_1 & + & 3y_2 & + & 3y_3 & \geq & 5 \\ 3y_1 & + & 8y_2 & + & 5y_3 & \geq & 3 \\ & & & & y_1, y_2, y_3 & \geq & 0 \end{array}$$

minimize the right-hand side of (2).
dual problem:

$$\begin{array}{lrcrcrcl} \text{minimize} & y_1 & + & 55y_2 & + & 3y_3 & & \\ \text{subject to} & y_1 & + & 5y_2 & - & y_3 & \geq & 4 \\ & -y_1 & + & y_2 & + & 2y_3 & \geq & 1 \\ & -y_1 & + & 3y_2 & + & 3y_3 & \geq & 5 \\ & 3y_1 & + & 8y_2 & + & 5y_3 & \geq & 3 \\ & & & & & y_1, y_2, y_3 & \geq & 0 \end{array}$$

Duality

primal problem.

$$\begin{array}{lrcll} \text{maximize} & \sum_{j=1}^{n} c_j x_j & & & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & i = 1, \ldots, m \\ & x_j & \geq & 0 & j = 1, \ldots, n \end{array}$$

associated dual problem

$$\begin{array}{lrcll} \text{minimize} & \sum_{i=1}^{m} b_j y_i & & & \\ \text{subject to} & \sum_{i=1}^{m} a_{ij} y_i & \geq & c_j & j = 1, \ldots, n \\ & y_i & \geq & 0 & i = 1, \ldots, m \end{array}$$

Weak duality

For every primal feasible solution $(x_1, \ldots, x_n)$
for every dual feasible solution $(y_1, \ldots, y_m)$:

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i$$

Proof

$$\sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \leq \sum_{i=1}^{m} b_i y_i$$

Deriving bounds efficiently

- At each branching node we add one constraint
- New LP-problems needs to be solved
- Can we reuse some computations ?

$$\begin{array}{ll} \text{maximize} & cx \\ \text{subject to} & Ax \leq b \\ & a'x \leq b' \\ & x \geq 0 \end{array}$$

- The previous solution is not feasible!
- Simplex needs feasible solutions in every step
- Consider dual problem

$$\begin{array}{ll} \text{minimize} & yb + y'b' \\ \text{subject to} & yA + y'a' \geq c \\ & y, y' \geq 0 \end{array}$$

- When primal problem gets additional constraint $a'x \leq b'$ the dual problem gets one more variable y'
- The same y is feasible to dual problem ($y' = 0$)
- Same basis solution (for dual problem) can be used
- Normally, only a few steps are needed to find new LP-optimum

CPLEX

The strategy for solving MIP problems

- Preprocessing
- Additional constraints
- Best-first search in branch-and-bound (Space consuming)
- Dual simplex in every iteration
- Branch on variables close to integer solution
- Then branch on other variables ("the mess")
- CPLEX only finds approximate solutions to MIP problems

The use of interior-point algorithms

- Simplex runs in exponential time (worst-case)
- Interior-point algorithms solve LP-problem in polynomial time
- May be useful for solving MIP problems, if degenerate problem
- Use interior-point to find LP-relaxation at root node
- Derive dual solution (Complementary slackness)
- Use dual simplex at other branching nodes

Solving IP models — hierarchy of techniques

Some IP naturally lead to integer solutions

- Totally unimodular matrices
- Several transportations problems and network problems are totally unimodular.

Preprocessing and reformulation

- Reformulation of constraints to TU
- Tightening bounds
- Variable fixing
- Redundant constraints

Branch-and-bound methods

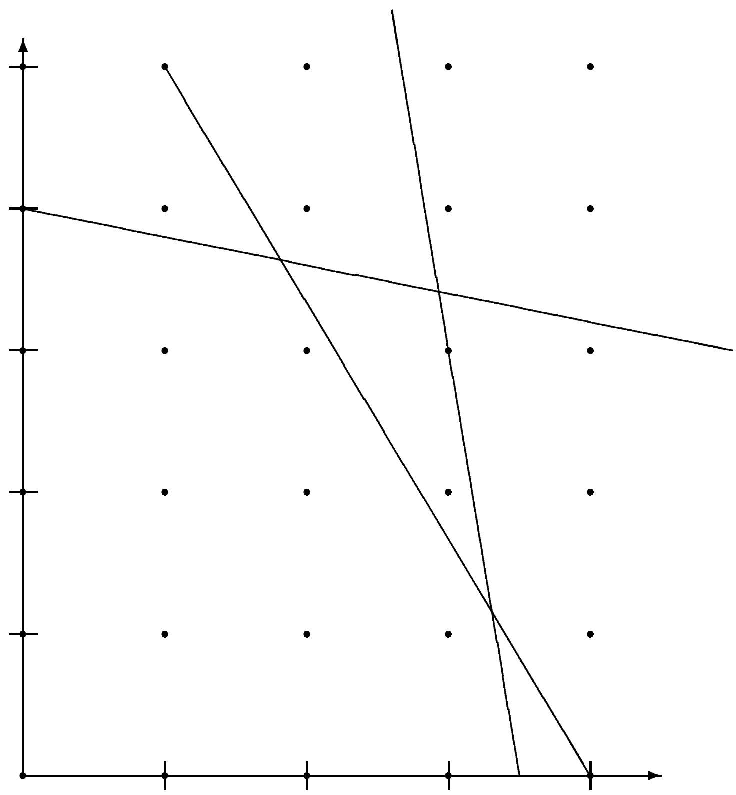
- Branching strategy
- Dual simplex

Preprocessing

LP: reduce number of constraints (LP-redundant), tighten bounds

IP: extend number of constraints (IP-redundant), closer to convex hull

Since the model will be solved several times during branch-and-bound, preprocessing will pay off.



Preprocessing

Example:

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & + & x_2 & - & x_3 & & \\
\text{subject to} & 5x_1 & - & 2x_2 & + & 8x_3 & \leq & 15 \\
 & 8x_1 & + & 3x_2 & - & x_3 & \geq & 9 \\
 & x_1 & + & x_2 & + & x_3 & \leq & 6 \\
 & 0 & \leq & x_1 & \leq & 3 & & \\
 & 0 & \leq & x_2 & \leq & 1 & & \\
 & 1 & \leq & x_3 & & & &
\end{array}
$$

Tightening bounds

Isolating x_1 in the first constraint

$$5x_1 \leq 15 + 2x_2 - 8x_3 \leq 15 + 2 - 8 = 9$$

thus $x_1 \leq \frac{9}{5}$.
Isolating x_3 in the first constraint

$$8x_3 \leq 15 + 2x_2 - 5x_1 \leq 15 + 2 - 0 = 17$$

thus $x_3 \leq \frac{17}{8}$.

In this way, pass through all variables, all constraints. Time complexity $O(mn^2)$. May be repeated several times.

Preprocessing, tightening bounds

$$
\begin{array}{ll}
\text{maximize} & \ldots \\
\text{subject to} & a_0 x_0 + \Sigma_{j=1}^{n} a_j x_j \leq b \\
 & \ell_j \leq x_j \leq u_j
\end{array}
$$

Then we have the additional bounds (which may be tighter)

- If $a_0 > 0$ then

$$x_0 \leq \frac{1}{a_0}\left(b - \sum_{j:a_j>0} a_j \ell_j - \sum_{j:a_j<0} a_j u_j\right)$$

- If $a_0 < 0$ then

$$x_0 \geq \frac{1}{a_0}\left(b - \sum_{j:a_j>0} a_j \ell_j - \sum_{j:a_j<0} a_j u_j\right)$$

Preprocessing, redundant constraints

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & + & x_2 & - & x_3 & & \\
\text{subject to} & 5x_1 & - & 2x_2 & + & 8x_3 & \leq & 15 \\
& 8x_1 & + & 3x_2 & - & x_3 & \geq & 9 \\
& x_1 & + & x_2 & + & x_3 & \leq & 6 \\
& \frac{7}{8} & \leq & x_1 & \leq & \frac{9}{5} & & \\
& 0 & \leq & x_2 & \leq & 1 & & \\
& 1 & \leq & x_3 & \leq & \frac{101}{64} & &
\end{array}
$$

Consindering constraint 3 we get

$$x_1 + x_2 + x_3 \leq \frac{9}{5} + 1 + \frac{101}{64} < 6$$

thus the constraint is redundant

Preprocessing, redundant constraints

$$\begin{array}{ll} \text{maximize} & \ldots \\ \text{subject to} & a_0 x_0 + \Sigma_{j=1}^n a_j x_j \le b \\ & \ell_j \le x_j \le u_j \end{array}$$

The constraint $a_0 x_0 + \Sigma_{j=1}^n a_j x_j \le b$ is *redundant* if

$$\sum_{j:a_j>0} a_j u_j + \sum_{j:a_j<0} a_j \ell_j \le b$$

The problem is *infeasible* if

$$\sum_{j:a_j>0} a_j \ell_j + \sum_{j:a_j<0} a_j u_j > b$$

Preprocessing, integer variables

$$\begin{array}{lrcrcrcrcr}
\text{maximize} & \ldots \\
\text{subject to} & 7x_1 & + & 3x_2 & - & 4x_3 & - & 2x_4 & \le & 1 \\
& -2x_1 & + & 7x_2 & + & 3x_3 & + & 4x_4 & \le & 6 \\
& & - & 2x_2 & - & 3x_3 & - & 6x_4 & \le & -5 \\
& 3x_1 & & & - & 2x_3 & & & \ge & -1 \\
& x & \in & \mathbb{B}^4
\end{array}$$

Generating logical inequalities

From constraint 1 we see that

- if $x_1 = 1$ then $x_3 = 1$, thus $x_1 \le x_3$
- if $x_1 = 1$ then $x_4 = 1$, thus $x_1 \le x_4$
- if $x_1 = 1$ and $x_2 = 1$ then infeasible, thus $x_1 + x_2 \le 1$

From constraint 2 we see that

- if $x_2 = 1$ then $x_1 = 1$, thus $x_2 \le x_1$

Combining inequalities

We have $x_1 + x_2 \le 1$ and $x_2 \le x_1$ thus $x_2 = 0$.

Simplifying

Insert $x_2 = 0$ in the model and repeat.

Preprocessing, integer variables

Variable fixation (maximization problem)

- For each variable x
- Consider the possible values of x as if we ran branch-and-bound algorithm starting with variable x
- If a branch has $\overline{z} \leq z$, drop this branch

Preprocessing, integer variables

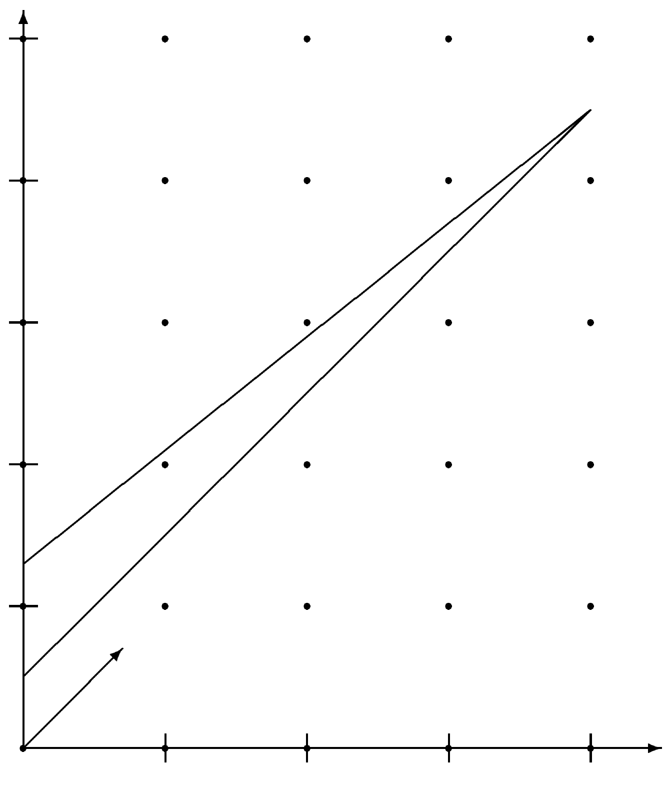
$$
\begin{array}{lrcrcrcrcr}
\text{maximize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & 7x_1 & + & 3x_2 & - & 4x_3 & - & 2x_4 & \leq & 1 \\
 & -2x_1 & + & 7x_2 & + & 3x_3 & + & 4x_4 & \leq & 6 \\
 & & - & 2x_2 & - & 3x_3 & - & 6x_4 & \leq & -5 \\
 & 3x_1 & & & - & 2x_3 & & & \geq & -1 \\
 & x & \in & \mathbb{B}^4 & & & & & &
\end{array}
$$

Assume that a solution with $z = 3$ has been guessed

- root, upper bound 3.29091
- $x_1 = 0$, upper bound 2
- $x_1 = 1$, upper bound 3
- $x_2 = 0$, upper bound 3
- $x_2 = 1$, infeasible
- $x_3 = 0$, upper bound 1.85454
- $x_3 = 1$, upper bound 3.29091
- $x_4 = 0$, infeasible
- $x_4 = 1$, upper bound 3.29091

Preprocessing

$$\begin{array}{ll} \text{maximize} & x_1 + x_2 \\ \text{subject to} & -2x_1 + \quad 2x_2 \geq 1 \\ & -8x_1 + 10x_2 \leq 13 \\ & x_1, x_2 \geq 0, \text{integer} \end{array}$$



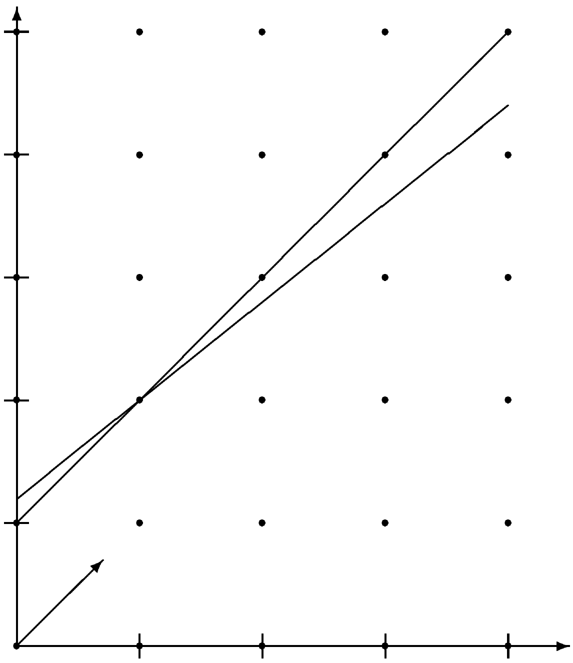
Preprocessing

$$\begin{array}{ll} \text{maximize} & x_1 + x_2 \\ \text{subject to} & -2x_1 + \quad 2x_2 \geq 1 \\ & -8x_1 + 10x_2 \leq 13 \\ & x_1, x_2 \geq 0, \text{integer} \end{array}$$

preprocess:

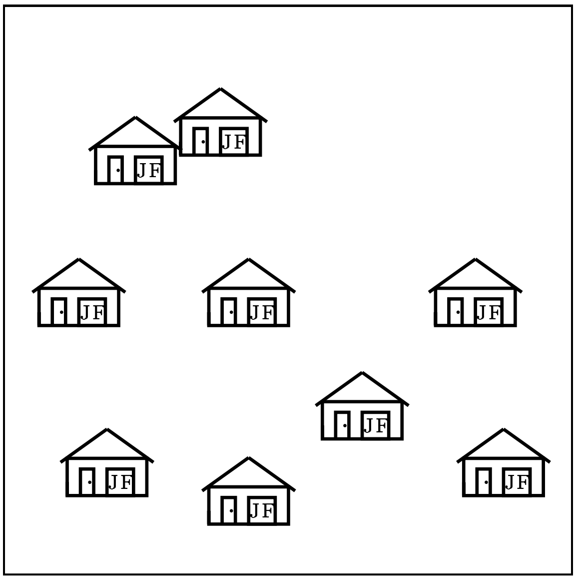
$$\begin{array}{rcl} -2x_1 + 2x_2 & \geq & 1 \\ -x_1 + x_2 & \geq & 1/2 \\ -x_1 + x_2 & \geq & 1 \end{array}$$

$$\begin{array}{rcl} -8x_1 + 10x_2 & \leq & 13 \\ -4x_1 + 5x_2 & \leq & 13/2 \\ -4x_1 + 5x_2 & \leq & 6 \end{array}$$



Example: A location problem

Dispersion problem: Open p out of n possible facilities so that their overall distance is maximized



- Distance i to j is $d_{ij} \geq 0$.
- $d_{ij} = d_{ji}$ and $d_{jj} = 0$.
- Binary variable x_j is one if facility open

p-dispersion problem

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} \sum_{i=1}^{n} d_{ij} x_i x_j \\
\text{subject to} \quad & \sum_{j=1}^{n} x_j = p \\
& x_j \in \{0,1\}, \quad j = 1, \ldots, n.
\end{aligned}
$$

p **dispersion problem, example**

Example:

$i \backslash j$	1	2	3	4	5	6	7
1	0	3	7	4	10	5	7
2	3	0	9	5	5	10	6
3	7	9	0	1	3	2	4
4	4	5	1	0	1	9	1
5	10	5	3	1	0	3	2
6	5	10	2	9	3	0	3
7	7	6	4	1	2	3	0

$n = 7$, $p = 3$.

Optimal solution is $x_2 = x_4 = x_6 = 1$, objective 48.

Linear formulation

Quadratic model

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} \sum_{i=1}^{n} d_{ij} x_i x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} x_j = p \\ & x_j \in \{0,1\} \end{aligned}$$

Introduce $y_{ij} = 1 \Leftrightarrow (x_i = 1 \text{ and } x_j = 1)$.

$$y_{ij} \le x_i, \qquad y_{ij} \le x_j, \qquad x_i + x_j \le 1 + y_{ij}$$

Linear model

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} y_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{n} x_j = p \\ & y_{ij} \le x_i && j = 1, \ldots, n \\ & y_{ij} \le x_j && i = 1, \ldots, n \\ & x_i + x_j \le 1 + y_{ij} && i, j = 1, \ldots, n \\ & x_j, y_{ij} \in \{0,1\} \end{aligned}$$

Constraint $y_{ij} = 1 \Leftarrow (x_i = 1 \text{ and } x_j = 1)$ not necessary.

Better linear formulation

Quadratic model

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} \sum_{i=1}^{n} d_{ij} x_i x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} x_j = p \\ & x_j \in \{0,1\} \end{aligned}$$

Constraint $y_{ij} = 1 \Leftarrow (x_i = 1 \text{ and } x_j = 1)$ not necessary.

Introduce $(y_{ij} = 1 \Rightarrow x_j = 1)$ and $(y_{ij} = 1 \Leftrightarrow y_{ji} = 1)$

$$y_{ij} \le x_j, \qquad y_{ij} = y_{ji},$$

Multiply $\sum_{i=1}^{n} x_i = p$ by x_j for each j getting

$$\sum_{i=1}^{n} x_i x_j = \sum_{i=1}^{n} y_{ij} = p x_j \qquad j = 1, \ldots, n$$

Linear model

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} y_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{n} x_j = p \\ & \sum_{i=1}^{n} y_{ij} = p x_j && j = 1, \ldots, n \\ & y_{ij} = y_{ji} && i, j = 1, \ldots, n \\ & y_{ij} \le x_j && i, j = 1, \ldots, n \\ & x_j, y_{ij} \in \{0,1\} \end{aligned}$$

Relaxation: drop $y_{ij} = y_{ji}$

Better linear formulation

Relaxed linear model

$$
\begin{array}{lll}
\text{maximize} & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} y_{ij} & \\
\text{subject to} & \sum_{j=1}^{n} x_j = p & \\
& \sum_{i=1}^{n} y_{ij} = p x_j & j = 1, \ldots, n \\
& y_{ij} \leq x_j & i, j = 1, \ldots, n \\
& x_j, y_{ij} \in \{0, 1\} &
\end{array}
$$

$$
\begin{array}{lccccccccccr}
\max & \boxed{\sum_{i=1}^{n} d_{i1} y_{i1}} & + & \boxed{\sum_{i=1}^{n} d_{i2} y_{i2}} & + & \boxed{\sum_{i=1}^{n} p_{i3} y_{i3}} & + \ldots + & 0x_1 & + 0x_2 & + 0x_3 & + \ldots & \\
\text{s.t.} & \boxed{\sum_{i=1}^{n} y_{i1}} & & & & & & -px_1 & & & & = 0 \\
& & & \boxed{\sum_{i=1}^{n} y_{i2}} & & & & & -px_2 & & & = 0 \\
& & & & & \boxed{\sum_{i=1}^{n} y_{i3}} & & & & -px_3 & & = 0 \\
& & & & & & & x_1 & +x_2 & +x_3 & + \ldots & = p
\end{array}
$$

p dispersion problem, deriving the bound

i \ j	1	2	3	4	5	6	7
1	0	3	7	4	10	5	7
2	3	0	9	5	5	10	6
3	7	9	0	1	3	2	4
4	4	5	1	0	1	9	1
5	10	5	3	1	0	3	2
6	5	10	2	9	3	0	3
7	7	6	4	1	2	3	0

$n = 7$, $p = 3$.

$d'_1 = 24$ $d'_2 = 25$ $d'_3 = 20$ $d'_4 = 18$ $d'_5 = 18$ $d'_6 = 24$ $d'_7 = 17$

Upper bound d'_j on each facility j

$$\begin{aligned} \text{maximize} \quad & d'_j = \sum_{i=1}^{n} d_{ij} y_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{n} y_{ij} = p \\ & y_{ij} \in \{0,1\}, \quad i = 1, \ldots, n. \end{aligned}$$

Upper bound $\overline{z}$

$$\begin{aligned} \text{maximize} \quad & \overline{z} = \sum_{j=1}^{n} d'_j x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} x_j = p \\ & x_j \in \{0,1\}, \quad j = 1, \ldots, n. \end{aligned}$$

Branch-and-bound algorithm

- Depth first search (bounds in $O(n)$ time)
- Branch by splitting on x_j
- Fixed order of variables according to d'_j
- Branch $x_j = 0$: remove row, column j
- Branch $x_j = 1$: decrease p, add d_{jj} to objective, set $d_{ii} := d_{ii} + d_{ij} + d_{ji}$ for $i = j+1, \ldots, n$. remove row, column j.

Branch-and-bound tests

GEO *geometrical problems*
d_{ij} euclidean distance between i and j

WGEO *weighted geometrical problems*
Each facility has a weight. d_{ij} euclidean distance between i and j times weights

EXP *exponential distribution*
d_{ij} with $i < j$ is randomly drawn from exponential distribution.

AEXP *asymmetric exponential distribution*
as above but $d_{ij} \neq d_{ji}$

RAN *random distances*
d_{ij} randomly distributed in $[1 \ldots 100]$.

DSUB *dense subgraph*
d_{ij} is set to 1 or 0 with 50% probability.

Branch-and-bound results

n	GEO	WGEO	EXP	AEXP	RAN	DSUB
10	7.66	6.76	7.90	7.57	7.68	8.30
20	9.04	4.03	11.11	11.60	11.94	16.37
30	9.15	3.25	14.84	13.40	12.88	16.58
40	8.59	1.47	14.97	13.45	8.64	11.24
50	9.07	3.39	16.17	9.77	20.89	27.86
60	19.69	2.68	22.31	15.69	15.52	—
70	10.80	3.20	—	—	—	—
80	6.93	2.76	—	—	—	—
90	—	2.92	—	—	—	—
100	—	7.09	—	—	—	—
150	—	2.38	—	—	—	—
200	—	2.78	—	—	—	—

Table 1: Relative deviation of upper bound $\overline{z}$ in pct. Average of 10 instances.

n	GEO	WGEO	EXP	AEXP	RAN	DSUB
10	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00
30	0.01	0.01	0.01	0.01	0.02	0.02
40	0.05	0.01	0.11	0.31	0.57	1.60
50	0.64	0.03	2.79	0.58	12.65	30.47
60	2.85	0.05	87.28	61.52	4552.81	—
70	39.18	0.09	—	—	—	—
80	153.15	0.17	—	—	—	—
90	—	0.33	—	—	—	—
100	—	0.44	—	—	—	—
150	—	3.08	—	—	—	—
200	—	161.21	—	—	—	—

Table 2: Solution times in seconds as average of 10 instances.

n	GEO	WGEO	EXP	AEXP	RAN	DSUB
10	12	7	8	7	9	6
20	140	18	81	227	328	448
30	1654	45	2082	2659	5716	8420
40	12675	26	42851	141162	276980	927312
50	220355	858	1105817	218292	5565562	16162737
60	816524	554	28536918	19629045	3217643	—
70	9727736	1241	—	—	—	—
80	28711239	7282	—	—	—	—
90	—	16652	—	—	—	—
100	—	13646	—	—	—	—
150	—	123478	—	—	—	—
200	—	7302184	—	—	—	—

Table 3: Number of branch-and-bound nodes. Average of 10 instances.