Linear and Integer Optimization

Exercise Sheet 5

Exercise 5.1: (Dual Simplex Algorithm)
Let $\max\{c^\intercal x : Ax = b, x \geq 0\}$ be an LP that is not unbounded and where A has full row rank.
Consider the following algorithm that has as an input the LP and a dual feasible basis B, i.e. $z_N = c_N - A_N^\intercal A_B^{-\intercal} c_B \leq 0$.

BTRAN:
Solve $A_B x_B = b$

PRICING:
If $x_B \geq 0$, **Stop**. Else choose an $i \in \{1, \ldots, m\}$ mit $x_{B_i} < 0$.

FTRAN:
Solve $A_B^\intercal w = e_i$ and compute $\alpha_N = A_N^\intercal w$.

RATIO-Test:
If $\alpha_N \geq 0$, **Stop**. Else choose a

$$j = \arg\min \left\{ \frac{z_k}{\alpha_k} : \alpha_k < 0, k \in N \right\}, \text{ and set } \gamma = \frac{z_j}{\alpha_j}.$$

Update:

$$\begin{aligned} z_N &\leftarrow z_N - \gamma \alpha_N, & z_{B_i} &\leftarrow -\gamma, \\ N &\leftarrow N \setminus \{j\} \cup \{B_i\}, & B_i &\leftarrow j \quad (\text{now } B \leftarrow B \setminus \{B_i\} \cup \{j\}). \end{aligned}$$

Goto **BTRAN**.

Prove

1. If the algorithm stops in the PRICING step, then B is an optimum basis and x_B with $x_N = 0$ is an optimum basic solution. (2 Points)
2. If the algorithm stops in the RATIO-Test, then $P^=(A, b) = \emptyset$. (2 Points)
3. The UPDATE-step transforms a dual feasible basis B into a new basis that is dual feasible again. (3 Points)

Exercise 5.2: Consider the following linear program

$$
\begin{array}{rrcrcrcrcrcrcl}
\max & 9x_1 & + & 3x_2 & + & x_3 & & & & & & & & \\
\text{s.d.} & x_1 & & & & & + & x_4 & & & & & = & 1 \\
 & 6x_1 & + & x_2 & & & & & + & x_5 & & & = & 9 \\
 & 18x_1 & + & 6x_2 & + & x_3 & & & & & + & x_6 & = & 81 \\
 & x_1 & , & x_2 & , & x_3 & , & x_4 & , & x_5 & , & x_6 & \geq & 0
\end{array}
$$

Show that the simplex algorithm traverses all vertices of the polyhedron when using $(4,5,6)$ as a starting basis, and choosing $j \in \arg\max\{z_j \ : \ j \in N\}$ in the pricing step. (4 Points)

Exercise 5.3: (Job Assignment Problem)
In the job assignment problem, n jobs with execution times of $t_1, \ldots, t_n \in \mathbb{R}_+$ need to be processed by m workers. However not every worker can perform every job. For each job $i \in \{1, \ldots, n\}$ a set $S_i \subseteq \{1, \ldots, m\}$ specifies the workers that can perform job i. Several workers can process a job in parallel to speed up the processing time, but each worker can only work at a single job at the same time.

1. Formulate an LP minimizing the time until the last job is completed.
2. Determine the dual LP.
3. Develop a simple polynomial time algorithm for $n = 2$ and $t_1, t_2 > 0$ that finds a primal and dual optimal solution and prove its correctness.

(5 Points)

Submission deadline: Thursday, November 16, 2017, before the lecture (in groups of 2 students).
Programming exercise 2:
Implement the (revised) simplex algorithm from the lecture. The algorithm will not get an initial solution but has to compute a starting basis by itself.
Your program should decide if an instance is infeasible, unbounded or can be solved optimally and should return a vector proving infeasibility or unboundedness in the first two cases.
If an instance can be solved optimally, your program should output optimum primal and dual solutions as well as the corresponding objective function value.
That way, correctness of the output can easily be verified. You may choose an index-strategy by yourself.

As input, the program expects a text-file obeying the specification described in Programming Exercise 1 on Exercise Sheet 2. Instances and the input reader can be reused.

(20 Points)

If you implement the algorithm with rational numbers (instead of floating points), e.g. with the `gmp` library (gmplib.org), you can achieve 5 bonus points.

Submission of the programming exercise until Thursday, 07.12.2017, before the lecture via e-mail to your tutor and to held@or.uni-bonn.de!