Winter semester 2013/14
Prof. Dr. S. Held
D. Rotter, M. Sc.

Research Institute for Discrete Mathematics
University of Bonn

Linear and Integer Optimization

Exercise Sheet 6

Exercise 6.1: Consider the following LP with only one restricting equality:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} c_i x_i & \\ \text{s.t.} \quad & \sum_{i=1}^{n} a_i x_i = b & i = 1, \dots, n \\ & 0 \leq x_i \leq 1 & i = 1, \dots, n. \end{aligned}$$

1. Provide a simple feasibility test for the problem.
2. Give an algorithm with running-time $\mathcal{O}(n \log n)$ that finds an optimal solution.

(5 Points)

Exercise 6.2: Let $G = (V, E)$ be a directed graph with edge-capacities $u : E \to \mathbb{K}_+$ and let $s, t \in V(G)$ be two special vertices. Furthermore, let

$$\mathcal{P} := \{P \subseteq E : P \text{ is the edge-set of a } s\text{-}t\text{-path in } G\}.$$

Consider the following LP (P):

$$\begin{aligned} \max \quad & \sum_{P \in \mathcal{P}} y_P & \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}: e \in P} y_P \leq u(e) & \text{for all } e \in E \\ & y_P \geq 0 & \text{for all } P \in \mathcal{P}. \end{aligned}$$

1. Determine the dual (D) of (P) and give graph theoretical interpretations of (D) and (P).
2. Find a class of graphs for which the number of paths $|\mathcal{P}|$ is not polynomially bounded by $|V| + |E|$.

3. Formulate an equivalent program to (P) for which the number of inequalities is polynomially bounded by $|V| + |E|$.

(5 Points)

Exercise 6.3:
In the Network-Simplex, the fundamental circuit C of an edge $e \in E(G) \setminus T$ has to be computed in each iteration. If we have stored a pointer to the predecessor of v on the r-v-path in T for each vertex $v \in V(G)$, C can easily be determined in $\mathcal{O}(|V(G)|)$ time. On the other hand, $|V(G)| \gg |V(C)|$ holds for a lot of applications.

Show how the apex of C can be found ...

1. by traversing at most $2|V(C)|$ edges using the pointers to the predecessors and at most one additional memory-bit for each vertex.
2. by traversing at most $|V(C)|$ edges if in addition, the number of edges in the r-v-path in T is stored for each vertex v.

(5 Points)

Submission deadline: Tuesday, 26.11.2013, before the lecture.

Programming exercise 2:
Implement the revised Simplex Algorithm (Algorithm 3). 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 3. Instances and the input reader can be reused.

(16 Points)

If you implement the algorithm with rational numbers (instead of floating points), e.g. with the `gmp` library (gmplib.org), you will get 4 bonus points.

Submission of the programming exercise until Tuesday, 10.12.2013, before the lecture via e-mail to your tutor and to held@or.uni-bonn.de!