Winter semester 2017/18 Research Institute for Discrete Mathematics
Prof. Dr. S. Held University of Bonn
Dr. U. Brenner

Linear and Integer Optimization

Exercise Sheet 10

Exercise 10.1: Show that for each $K \in \mathbb{N}$ there is a bounded integer program with two variables such that the branch-&-bound algorithm visits more than K vertices in the branch-&-bound tree. (4 Points).

Exercise 10.2: Consider the following ILP (integer linear program)

$$\begin{array}{ll} \max -\sqrt{2}x + y & \\ -\sqrt{2}x + y & \leq 0 \\ x & \geq 1 \\ y & \geq 0 \\ x, y & \in \mathbb{Z} \end{array}$$

Show:

1. The ILP has feasible solutions. (1 Point)
2. The objective is bounded from above by 0. (1 Point)
3. There is no optimum solution. (2 Points)
4. Let S be the set of solutions of the ILP, then $\mathrm{conv}(S)$ is not a polyhedron. (1 Point)

Exercise 10.3: (Mixed integer programming hull)
Let $P = \{x \in \mathbb{R}^{k+l} : Ax \leq b\}$ be a rational polyhedron. Show that $\mathrm{conv}(P \cap (\mathbb{Z}^k \times \mathbb{R}^l))$ is a rational polyhedron. (5 Points)

Exercise 10.4:
The Maximum-Stable-Set-Problem is defined as follows. Given a graph G, we are looking for a vertex set $S \subseteq V(G)$ of maximum cardinality $|S|$ such that $\{v, w\} \notin E(G)$

for all $v, w \in S$. A vertex set S with $E(G[S]) = \emptyset$ is called *stable set*. This problem can be modeled by the following ILP:

$$\max \sum_{v \in V(G)} x_v \tag{1}$$

$$s.d. \quad x_v + x_w \leq 1 \qquad \forall \{v, w\} \in E(G) \tag{2}$$

$$x_v \in \{0, 1\} \qquad \forall v \in V(G) \tag{3}$$

Show that following inequalities are valid for the Maximum-Stable-Set-Problem

$$\sum_{v \in V(H)} x_v \leq \left\lceil \frac{|V(H)| - 1}{2} \right\rceil \quad \text{for a circuit } H \subseteq G.$$

Give examples, where the optima of the linear relaxations are reduced (strictly). (2 Points)

Submission deadline: Thursday, December 21, 2017, before the lecture (in groups of 2 students).

Programming Exercise 3:
Implement the branch-&-bound algorithm for the MAXIMUM-WEIGHT-STABLE-SET-PROBLEM that is defined as follows. Given a graph G and weights on the vertices $\alpha : V(G) \to \mathbb{N}$, we are looking for a stable set $S \subseteq V(G)$ of maximum weight $\sum_{v \in S} \alpha(v)$. It should be modeled by the following ILP:

$$\max \sum_{v \in V(G)} \alpha(v) x_v \tag{1}$$

$$s.d. \quad x_v + x_w \leq 1 \qquad \forall \{v, w\} \in E(G) \tag{2}$$

$$x_v \in \{0, 1\} \qquad \forall v \in V(G) \tag{3}$$

As an LP solver you must use the academically free program **QSopt** through the API in `lp.h` that is available on the web-site. To make the implementation easier, you find a program that

1. Reads an instance,
2. creates the above LP-relaxation using the API in `lp.h`,
3. solves it and prints the solution vector to the console.

(see `http://www.or.uni-bonn.de/~held/lpip/1718/mss.zip`). The ZIP file contains also test instances. **Read the README file for further information!**. The program compiles under Linux (for Windows/Cygwin, you may try `http://www.or.uni-bonn.de/~held/lpip/1314/mss.zip`). For compiling type 'make' in the extracted directory 'mss'. If you encounter problems building mss, do not hesitate to contact me: held@dm.uni-bonn.de. Note that the matrix A is usually sparse, i.e. most coefficients are zero. Thus, in `lp.h` new rows/constraints are always defined by their non-zero entries. The corresponding functions in `lp.h` are commented and their use becomes clear in mss.c.
As shown in the lecture, the LP relaxation has a large integrality gap, which is problematic for branch-&-bound. Thus you should first try to tighten the gap in the root LP, by adding clique inequalities and inequalities of Exercise 10.4. To this end you should implement a simple greedy algorithms. E.g. for cliques you can start with $C = \{v\}$ for a $v \in V(G)$ and add vertices $w \in V(G) \setminus C, C \subseteq \Gamma(w)$, with a large value x_w to C.
This should be started iteratively with different vertices $v \in V$ until all vertices are part of some (inclusion-wise) maximal clique. You should iterate the two steps

- solving the root lp and
- adding clique inequalities and circuit inequalities from Exercise 10.4

until no more clique inequalities are found before starting branch-&-bound. The algorithm should write

1. the value of the root LP without clique inequalities,
2. the value of the root LP with clique inequalities, and
3. the value and vertex indices of a maximum-weight stable set S

to the console. (20 Points)

Submission deadline of the programming exercise: Thursday, January 11, 2018, before the lecture.