Universität Bremen

FB3: Mathematik/Informatik

Dr. Martin Böhm
Dr. Ruben Hoeksma
Prof. Dr. Nicole Megow

Summer 2019

Approximation Algorithms

Exercise Sheet 3 (07. 05. 2019)

Original program:	*In the dual:*
maximum	minimum
$\max c^T x$	$\min b^T y$
m constraints n variables	m variables n constraints
the i-th constraint is $\leq$ the i-the constraint is $\geq$ the i-th constraint is $=$	$y_i \geq 0$ $y_i \leq 0$ $y_i \in \mathbb{R}$
$x_j \geq 0$ $x_j \leq 0$ $x_j \in \mathbb{R}$	the j-th constraint is $\geq$ the j-th constraint is $\leq$ the j-th constraint is $=$

Exercise 1. Dualize the following LP:

$$\begin{aligned} \max x_1 - 2x_2 + 3x_4 \\ x_2 \leq 0 \\ x_4 \geq 0 \\ x_2 - 6x_3 + x_4 \leq 4 \\ -x_1 + 3x_2 - 3x_3 = 0 \\ 6x_1 - 2x_2 + 2x_3 - 4x_4 \geq 5 \end{aligned}$$

Exercise 2. Dualize the linear programming relaxation of the integer program for Minimum vertex cover for a weighted graph $G = (V, E, w)$. To be precise, the task is to dualize the following:

$$\begin{aligned} \min \sum_{v \in V} w(v) x_v \\ \forall e = (uv) \in E : x_u + x_v \geq 1 \\ \forall v \in V : x_v \geq 0 \end{aligned}$$

In the following exercises we will investigate the possibility of using exponentially many constraints in an LP formulation. We must be careful here – if we allow arbitrary sets of exponentially many constraints (or exponentially many variables), we would quickly stray away from the useful theorem that LPs can be solved in polynomial time.

To guarantee that our LPs with exponentially many constraints can be solved, we need the following subroutine to exist:

Definition 1. *A* separation oracle *is an algorithm that satisfies the following:*

- *On input it receives any solution of an LP that is* guaranteed infeasible*.*
- *In polynomial time, it returns any inequality of the LP that is violated by the solution on input.*

Notice that the separation oracle does not have time to check all the exponentially many constraints, and must use some property of the LP to figure out which inequality is not holding.

Exercise 3. Formulate an LP for the MINIMUM WEIGHT SPANNING TREE problem. Use exponentially many constraints and argue what the separation oracle should do.

Exercise 4. Formulate an ILP (integer constraints are allowed) for MAXIMUM MATCHING in a graph.

Exercise 5. Formulate a linear program which solves the task SHORTEST s, t-PATH in an unweighted directed graph.

Explain your idea behind the LP, and then dualize it.

Exercise 6. Formulate an ILP for the TRAVELLING SALESMAN PROBLEM.