OPTIMIZATION METHODS: CLASS 10

Primal-dual algorithms

D: Suppose there is a (difficult) optimization problem with OPT as the value of the objective function. We say that an algorithm A is a *k-approximation algorithm* for this problem if A runs in polynomial time, returns a feasible solution on every input and the value of the objective function for any solution A produced by A is bounded by $A \leq k \cdot OPT$ (in the case of maximization, we want $OPT \leq k \cdot A$).

O:For every solution of a maximization integer LP and for its LP relaxation it holds that $OPT_{LP} \geq OPT_{ILP}$. In case of minimization, we have $OPT_{LP} \leq OPT_{ILP}$.

D(Slack**):** Suppose we have a system of linear inequalities (S) and, more specifically, the j-th inequality

$$a_{j1}x_1 + a_{j2}x_2 + a_{j3}x_3 + \ldots + a_{jn}x_n \leq b_j.$$

Suppose we are also given a vector x' that satisfies the j-th inequality. Then the *slack* of the j-th inequality and the solution x' is $s_j^{(S)} = b_j - \sum_{i=1}^{n} a_{ji}x'_i$.

Notice that it always holds that $s_j^{(S)} \geq 0$. If the inequality is $\geq$, we define the slack as $s_j^{(S)} = \sum_{i=1}^{n} a_{ji}x'_i - b_j$, so that again $s_j^{(S)} \geq 0$.

T(Complementary slackness**):** Assume we have a linear program (P) and its dual (D) of the following form.

$$\max c^T x, Ax \leq b, x \geq 0, \tag{P}$$
$$\min b^T y, A^T y \geq c, y \geq 0. \tag{D}$$

We are also given a pair of feasible solutions of the primal and dual (x', y'). Then the following holds: The pair (x', y') is a pair of optimal solutions if and only if all the following conditions are satisfied:

$$\forall i \in \{1, \ldots, n\} \colon x'_i \cdot s_i^{(D)} = 0, \tag{1}$$
$$\forall j \in \{1, \ldots, m\} \colon s_j^{(P)} \cdot y'_j = 0. \tag{2}$$

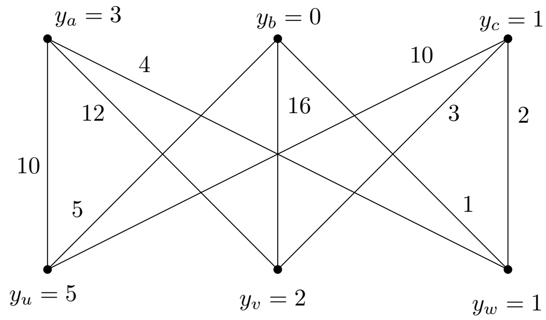
D:For a graph $G = (V, E)$ with two special vertices s, t, an *s, t-cut* is a subset of vertices C such that $s \in C, t \notin C$.

Exercise one   You have just been presented a 2-approximation algorithm for Weighted Vertex Cover, where we generated a pair of feasible solutions (x, y). This pair of feasible solution will not usually be an optimal pair – it is just a 2-approximation.

Check the complementary slackness conditions and explain which ones hold and which do not.

Exercise two   Below is a bipartite graph with weights on the edges. Next to each vertex you see a value to a supposedly optimal dual solution for Perfect Matching of Minimum Cost.

Prove that this dual solution is optimal.



EXERCISE THREE Formulate SHORTED s,t-PATH in a positive weighted undirected graph as an $\{0,1\}$-integer linear program. Your program should use exponentially many conditions, in fact, one for each s,t-cut in the graph. Dualize this program afterwards.

EXERCISE FOUR Consider the following algorithm:

1. $\vec{y} \leftarrow 0$, where y is a vector of dual variables.
2. $F \leftarrow \emptyset$
3. While there is no s,t-path in $G[F]$:
4. Consider the unique connected component C in $G[F]$ which contains s.
5. Increase y_C until some constraint (corresponding to an edge e) is tight.
6. Add e to F.
7. For each $e \in F$:
8. If $F \setminus \{e\}$ contains an s,t-path, remove e from F.
9. Return F as the shortest s,t-path.

Prove that this algorithm finds a shortest path.

EXERCISE FIVE MINIMUM STEINER FOREST (MSF) is the following problem: on input we get an undirected weighted graph $G = (V, E, w)$ with weights on the edges ($w : E \to \mathbb{R}^+$) and we also get a collection of disjoint sets $S_1, S_2, \ldots, S_k \subset V$. Your task is to find a set $F \subseteq E$ of minimum weight such that every two vertices $u, v \in S_i$ (for every i) belong to the same component in $G[F]$. $G[F]$ is clearly an acyclic graph – thus we call it a Steiner forest.

Formulate MSF as an integer program, write its relaxation, dualize such relaxation, and finally list the relevant complementary slackness conditions.

EXERCISE SIX Formulate a primal-dual algorithm for MSF. *Hint:* The algorithm should be similar to the one for the shortest path problem.
Voluntary homework: Argue that this algorithm is a 2-approximation algorithm.