Approximation Algorithms

Winter term 2021/22

Prof. Dr. Martin Hoefer
Tim Koglin, Lisa Wilhelmi

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institute of Computer Science
Algorithms and Complexity

Assignment 8

Issued: 14.12.2021
Due: 18.01.2022, **10:15h**

- This is the first assignment that counts for Part 2 of the course and is due in five weeks.
- Exercises marked with * are bonus - they count for your score but not for the sum of points.
- The next assignment will be released on Jan 18th, 2022.

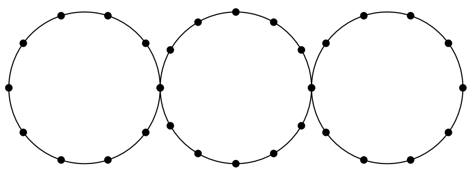
Exercise 8.1. *Dual LP* (3 + 3* points)

Consider the FEEDBACK VERTEX SET problem: the input is an undirected graph $G = (V, E)$ with weights $w_v \geq 0$, for each vertex $v \in V$. The goal is to determine a set of vertices $S \subseteq V$ with minimum total weight such that any simple cycle C of arbitrary length in G contains at least one vertex in S. Recall that a cycle is *simple* if only the first and last vertex are equal. For a given instance G, let $\mathcal{K}$ denote the set of all simple cycles in G. An LP-relaxation for FEEDBACK VERTEX SET is as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_{v \in V} w_v x_v, \\ \text{subject to} \quad & \sum_{v \in C} x_v \geq 1 \quad \forall C \in \mathcal{K}, \\ & x_v \geq 0 \quad \forall v \in V. \end{aligned}$$

a) State the associated dual LP and give a short interpretation of it.

b)* For the instance pictured below, assume that $w_v = 1$, for all $v \in V$. State three different feasible dual solutions $\boldsymbol{y}^{(1)}$, $\boldsymbol{y}^{(2)}$, $\boldsymbol{y}^{(3)}$ for this instance such that, for each solution, no higher value can be assigned to any dual variable separately. One of the three stated solutions shall be optimal – explain why it is optimal indeed.



Exercise 8.2. *LP for* SHORTEST *s-t*-PATH (4 points)

In the SHORTEST *s-t*-PATH problem, an undirected graph $G = (V, E)$ is given where each edge $e \in E$ is associated with a length $\ell_e \geq 0$. For two given vertices $s, t \in V$, the goal is to compute a path P that starts in s and ends in t with minimal sum of lengths. A set $U \subset V$ is called an *s-t-intersection* if $s \in U$ but $t \notin U$. Further, let $\delta(U)$ denote the set of all edges $\{u, v\} \in E$ such that $u \in U$ and $v \notin U$. The SHORTEST *s-t*-PATH problem can then be formulated as the following integer linear program (ILP):

$$\begin{aligned} \text{Minimize} \quad & \sum_{e \in E} \ell_e x_e, \\ \text{subject to} \quad & \sum_{e \in \delta(U)} x_e \geq 1 \qquad \forall U \in \mathcal{S}, \\ & x_e \in \{0, 1\} \quad \forall e \in E, \end{aligned}$$

where $\mathcal{S}$ is the set of all possible *s-t*-intersections, i.e., $\mathcal{S} = \{U \subset V \mid s \in U, t \notin U\}$.

For a set $A \subseteq E$ of edges, let $\mathbf{x}$ be the associated assignment of ILP variables where $x_e = 1$ for all $e \in A$, and $x_e = 0$ otherwise. Show that $\mathbf{x}$ is a solution to the ILP if and only if there exists an *s-t*-path P such that all edges in P are also contained in A.

Exercise 8.3. *Non-linear Rounding* (3 + 2 + 2 points)

As discussed in the lecture, the *Best-of-Rounding* algorithm for the MAXSAT problem runs *Simple rounding* and *LP rounding* once each and returns the solution that has higher total weight of the two. It was shown that *Best-of-Rounding* has approximation ratio 4/3. The goal of this exercise is to show that the same approximation ratio can be achieved without need for *Simple rounding* by solely rounding the solutions of the LP in a non-linear manner.

Using the same definitions and notations as in the lecture, let $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ denote an optimal solution to the LP-relaxation for MAXSAT with $k \in \mathbb{N}$ clauses. For starters, let $f : [0, 1] \to [0, 1]$ be an arbitrary function. The algorithm *LP f-rounding* first computes the optimal solution $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ and then independently sets $x_j = 1$ with probability $f(y_j^*)$ and $x_j = 0$ otherwise, for each variable x_j. Note that this is equivalent to the *LP rounding* algorithm from the lecture if $f(x) = x$. For the remainder of this exercise, *LP f-rounding* is considered for arbitrary functions f which satisfy the following property for all $x \in [0, 1]$:

$$1 - 4^{-x} \leq f(x) \leq 4^{x-1}.$$

Perform the following steps to show that *LP f-rounding* has approximation ratio 4/3 in expectation:

a) Show that any clause C_i is satisfied with probability at least $1 - 4^{-z_i^*}$.

b) Taking up the result in a), show that any clause C_i is satisfied with probability at least $3/4 \cdot z_i^*$. *Hint: Use the fact that the function* $g(x) = 1 - 4^{-x}$ *is concave in* $[0, 1]$ *to obtain a lower bound for* $g(x)$*, as done in the lecture.*

c) Let W_i be a random variable such that $W_i = w_i$ if C_i is satisfied and $W_i = 0$ otherwise, where w_i is the weight of clause C_i. Define $W = \sum_{i=1}^{k} W_i$ as the total weight of all satisfied clauses. Show that $\mathbb{E}[W] \geq 3/4 \cdot \text{OPT}$.

Exercise 8.4. *Randomized Algorithm for* Max3Sat (2 + 1 + 3 points)

The Max3Sat problem is a special case of the MaxSat problem discussed in the lecture. An instance of Max3Sat is given by $n \in \mathbb{N}$ Boolean variables $x_1, \ldots, x_n$ and $k \in \mathbb{N}$ clauses $C_1, \ldots, C_k$. Every clause C_i consists of exactly three literals and has weight $w_i = 1$. An assignment sets each variable to either true or false. The goal is to determine an assignment that maximizes the number of satisfied clauses.

a) Design a randomized algorithm for Max3Sat that computes an 8/7-approximate solution in expectation.

In the remainder of the exercise, the goal is to show that based on a) an approximation ratio of 8/7 can be achieved with certainty.

b) Show that there exists an assignment that satisfies at least $7/8 \cdot k$ clauses, for every instance of Max3Sat.

c) Show that any assignment drawn uniformly at random satisfies at least $7/8 \cdot k$ clauses with probability at least $1/(8k)$.
Hint: Use the probability p_l that exactly $l \in [k]$ clauses are satisfied.

Comment: Every application of the algorithm in a) yields a solution that satisfies $7/8 \cdot k$ clauses in expectation. Since such a solution exists for every instance, as shown in b), it is obtained with certainty by repeatedly applying the algorithm. The result in c) implies that the number of repetitions required is $8k$ in expectation.

We wish you a nice Christmas break
and a happy New Year!

The assignments and further information on the course are provided on our website:
https://algo.cs.uni-frankfurt.de/lehre/apx/winter2122/apx2122.shtml

Contacts: {koglin,wilhelmi}@em.uni-frankfurt.de.