Exam Optimization: Special Topics - aj 2015-2016

Date+Time: Tuesday, June 21, 2016, 09.00-13.00.

The exam is an "open book" exam. There are four questions. You can fill in the answers directly after the questions. This exam consists of 15 pages.

Hints:

- Read carefully!
- Write clearly!
- Mention your name!

Good Luck!

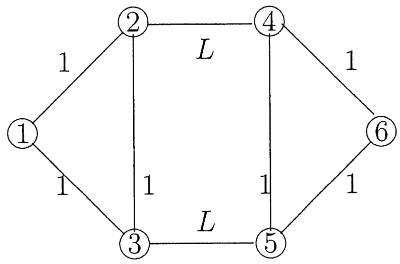
Question 1 (5 points)

About the Perfect Matching problem. Given is a set of persons $N = \{1, 2, \dots, n\}$, with n even, that needs to be paired up into $\frac{n}{2}$ pairs of persons. When distinct persons $i, j \in N$ are paired up, there is a given cost c_{ij} incurred. A perfect matching is a pairing of persons such that each person is matched up with exactly one other person. Of course, the goal is to find a perfect matching with minimum total cost. We represent this problem by a graph $G = (N, E)$ where each person is represented by a node, and the cost of an edge $e = \{i, j\}$ is $c_{i,j}$. Using a binary variable $x_{i,j}$ indicating whether persons i and j are paired up, we build the following integer programming formulation, referred to as (P):

$$
\begin{array}{lll}
\text{(P) Minimize} & \sum_{i,j \in N} c_{i,j} x_{i,j} & \\
\text{subject to} & \sum_{j \in N \setminus \{i\}} x_{i,j} = 1 & \text{for } i \in N, \\
 & x_{i,j} \in \{0, 1\} & \text{for all } i, j \in N.
\end{array}
$$

a. Is formulation (P) a correct formulation of the Perfect Matching problem? Explain your answer.

Consider an instance of the Perfect Matching problem depicted below, where the numbers corresponding to the edges are the corresponding cost-coefficients (and where L denotes a large positive number).



Figuur 1: An instance

b. What is the optimal solution value of the instance of Perfect Matching depicted in Figure 1? 2+L

c. What is the solution to the linear programming relaxation of formulation (P) for this instance? 3

d. Use integer rounding to derive a set of inequalities that are valid for the LP-relaxation of (P). Do they cut off the solution found in c?

e. Apply Lagrangian relaxation to (P) using Lagrangian multipliers $\lambda_i, i \in N$ for the constraints in (P). How does the resulting formulation look like? (Simplify the formulation as much as you can).

f. When given the λ_i's, how would you solve the resulting formulation?

Question 2 (5 points)

Consider the following problem. Given is an $n \times n$ matrix A with nonnegative entries $a_{i,j}$, $1 \leq i, j \leq n$. Moreover, the entries $a_{i,j}$ are such that, when summed over a column, and when summed over a row, the resulting sums equal 1, i.e., $\sum_{i=1}^{n} a_{i,j} = \sum_{j=1}^{n} a_{i,j} = 1$. Then, such a matrix A can be written as a convex combination of so-called *doubly stochastic* matrices.

Definition: An $n \times n$ matrix P is called doubly stochastic if all entries of P equal either 0 or 1, and if each row, as well as each column of P, contains a single 1.

Thus, it is a fact that we can write $A = \alpha_1 P_1 + \alpha_2 P_2 + \ldots \alpha_k P_k$, for some integer $k \geq 1$, with $\alpha_\ell > 0$ for each ℓ, and $\sum_\ell \alpha_\ell = 1$, and with $P_1, P_2, \ldots, P_k$ all doubly stochastic matrices. Our problem is to write the given matrix A as a convex combination of as few doubly stochastic matrices as possible. In other words, our problem is to minimize the number of doubly stochastic matrices needed to express A as a convex combination.

a. Write down an IP-formulation for this problem involving continuous variables α_ℓ, and involving a binary variable for each doubly stochastic matrix P_ℓ.

b. Give the LP-relaxation of this formulation. To what extent can you simplify it? Explain why it makes sense to apply column generation to this formulation.

c. What is the pricing problem? Explain your answer.

Question 3 (5 points)

Consider the following basic scheduling problem. Given are n jobs, each with a known processing time (or *length*) of p_j time units, $j = 1, \ldots, n$. Also given are m machines. The following optimization problem is of interest: assign each job to a machine so that the completion time of the job that is finished latest of all jobs (called the makespan) is as small as possible. We assume here that a job cannot be preempted, and can only be carried out by one machine; a machine can only process one job at a time. When, in addition, a deadline D is given, and we are interested in the question: does there exist a solution with deadline no more than D? we refer to the resulting decision problem as: `Makespan`.

a. Show that `Makespan` with $m = 2$ is at least as hard as Partition.

b. Show that `Makespan` with $m = 3$ is at least as hard as Partition.

c. Argue that the value of an optimal solution to our problem, referred to as OPT, satisfies the following inequality:

$$OPT \geq \max(\frac{\sum_{j=1}^{n} p_j}{m}, \max_{j=1,\dots,n} p_j). \quad (1)$$

Let us refer to the sum of processing times of jobs assigned to some machine as the *load* of that machine. Consider now the following algorithm for our problem, called `LIST`:

Choose any ordering of the jobs, and repeatedly assign each job to the machine with the current smallest load (and if there is a tie, choose a machine arbitrarily).

d. Show that `LIST` is a 2-approximation algorithm for our problem.

Question 4 (5 points)

TRUE or FALSE?? Please justify your answer shortly.

F a. Given a graph $G = (V, E)$, let s and t be two given nodes in V. A path from s to t that visits each other node in V precisely once is called a Hamiltonian $s - t$ path. Finding a shortest Hamiltonian $s - t$ path is a special case of finding a shortest path from s to t.

T b. The Assignment Problem is a special case of Min-Cost Flow.

T c. Partition is a special case of Knapsack.

F d. The Knapsack Problem is a relaxation of Partition.

T e. Integer Programming is at least as hard as the Traveling Salesman Problem.