UBINET, Master 2 IFI Algorithms for telecommunications

Final Exam, November 2013

3 hours

Lecture notes allowed. Not computers, cellphones, books.

Instruction and comments: the points awarded for your answer will be based on the correctness of your answer as well as the clarity of the main steps in your reasoning. All proposed solutions must be proved. **All the exercises are independent.** *The points are indicated so you may adapt your effort.*

1 Algorithmics on graphs

Exercise 1 (2 points, 20 minutes) Consider the elementary network flow N depicted in Figure 1 (left) and the initial flow f from s to t in Figure 1 (right).

- What must be checked to show that f is a flow? What is the value of the flow f?
- Apply the Ford-Fulkerson Algorithm to N starting from the flow f. The first two steps (in particular, the auxiliary digraphs) of the execution of the algorithm must be detailed.
- Give the flow and the cut obtained. Conclusion (recall the min cut-max flow theorem)?

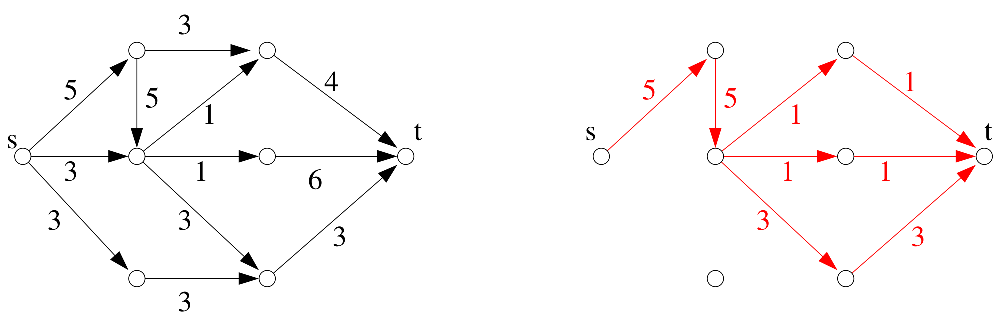


Figure 1: (left) Elementary network flow with arcs' capacity in black. (right) A flow f from s to t: a number close to an arc indicates the amount of flow along it. Arcs that are not represented have no flow.

Exercise 2 (3 points, 30 minutes) The goal of this exercise is to show an application of flows to organize the defense of the projects of some students.

Assume that the students $\{S_1, \cdots, S_n\}$ have to present their work to some professors at the end of their projects. There are q professors $\mathcal{P} = \{P_1, \cdots, P_q\}$. Each student S_i has a project Q_i, $i \leq n$.

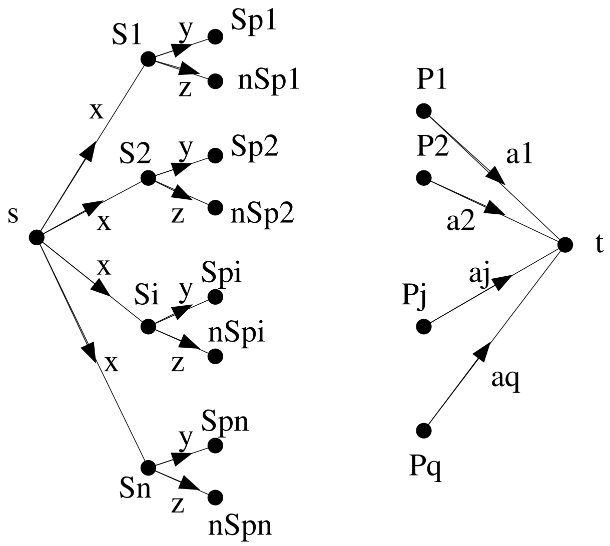
An assignment consists in organizing the juries that is determining which professor will attend which presentation.

However the professors have a constraint: each professor P_j, $j \leq q$, can attend at most a_j defenses. Furthermore we want to insure fairness between students. For any project, each professor is either a specialist of the subject or not. That is, for any $i \leq n$, $\mathcal{P}$ is partitionned into Sp_i and nSp_i, respectively the subset of the professors that are specialist of the project Q_i, and the professors that are not. The objective is that each student S_i present his work to x professors, y of them are specialists of Q_i and $z = x - y$ of them are not. An assignment is said to be feasible if it satisfies the constraint of the professors and the fairness issue for students.

We aim at using a flow-model to organize the juries (which professor will attend which presentation). For this purpose, we model the instance of the problem as a graph that is partially depicted below.

1. Explain how the figure below models the above problem. In particular, complete the graph by adding arcs between nodes Sp_i, nSp_i and the nodes P_j.
2. Consider any flow f from s to t. Explain how it define an assignment for the jury.

3. How can we decide whether satisfying all requirements of the problem is feasible?



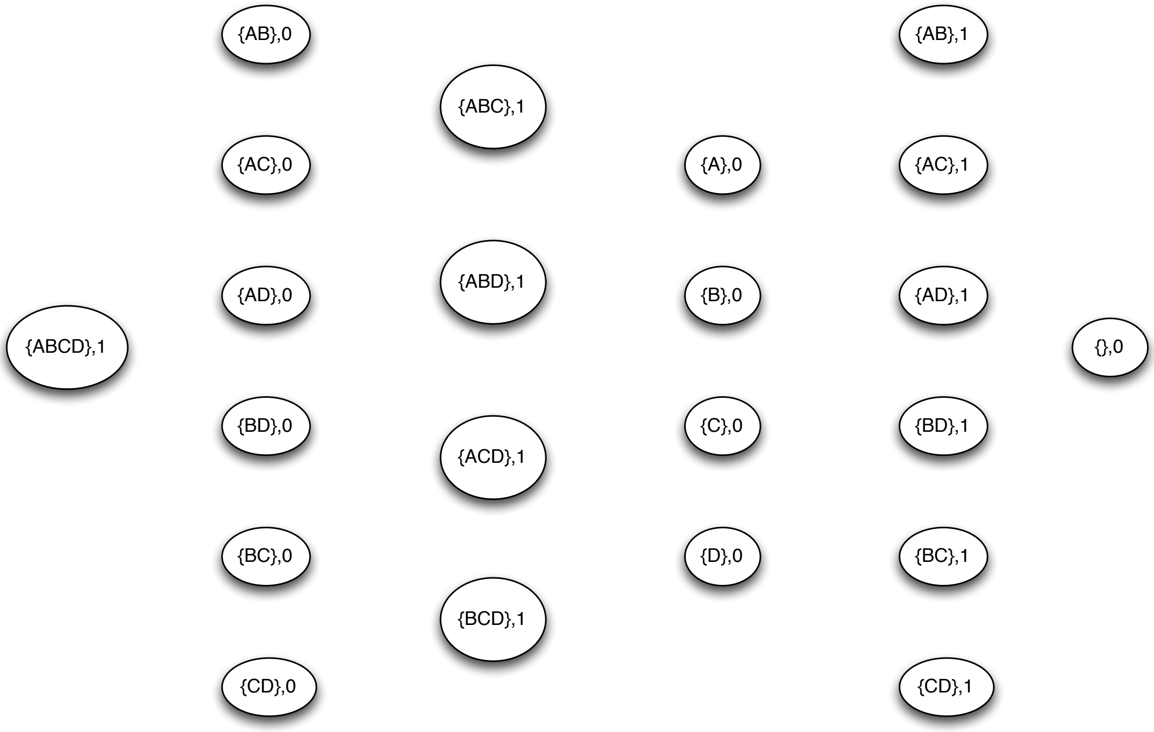
Exercise 3 (3 points, 30 minutes) Four imprudent walkers (let us call them A, B, C and D) are caught in the storm at night. To reach the shelter, they have to cross a canyon over a fragile rope bridge which can resist the weight of at most two persons. In addition, crossing the bridge requires to carry a torch to avoid to step into a hole. Unfortunately, the walkers have a unique torch and the canyon is too large to throw the torch across it. Due to dizziness and tiredness, the four walkers can cross the bridge in 1, 2, 5 and 10 minutes (for A, B, C and D respectively). When two walkers cross the bridge, they both need the torch and thus cross the bridge at the slowest of the two speeds.

Goal: The goal is to compute the minimum time for the walkers to cross the bridge. For this purpose, we model the problem with the following graph.

The graph model: Each node of the graph corresponds to the set of walkers that have not crossed the canyon yet and to the position of the torch. For instance, the node $(\{A, C\}, 0)$ corresponds to the configuration where walkers A and C are still on the bad side of the canyon and do not have the torch. The node $(\{B\}, 1)$ means that B is the only walker who has not crossed the canyon but he has the torch.

There is an arc from node u to v with length x if we can go from configuration u to configuration v by crossing the bridge once, in time x. For instance, initially, the configuration is $u = (\{A, B, C, D\}, 1)$ and A and C can cross the bridge in 5 minutes. Hence, there is an arc from u to $v = (\{B, D\}, 0)$ with length 5. That is, initially, A, B, C and D are on the bad side of the canyon with the torch, then A and C cross the canyon with the torch, which takes 5 minutes.

1. Draw the graph G for the given instance of the problem (to help you, Figure below represents the nodes of this graph).
2. Using an algorithm seen during the lecture, applied on G, explain how to compute the minimum time for the walkers to cross the bridge?
3. What is the minimum time for the walkers to cross the bridge? (this question is easy but may take a long time).



2 Linear Programming

Exercise 4 (2 points, 20 minutes) *We consider the following linear program.*

$$
\begin{array}{lrcrcrcrcr}
\textit{Minimize} & 8x_1 & + & 6x_2 & + & 7x_3 & + & 5x_4 & & \\
\textit{Subject to:} & & & & & & & & & \\
& 2x_1 & + & 6x_2 & + & 3x_3 & + & x_4 & \geq & 4 \\
& 4x_1 & + & 4x_2 & + & x_3 & + & 5x_4 & \geq & 2 \\
& 2x_1 & + & 2x_2 & + & x_3 & + & x_4 & \geq & -1 \\
& x_1 & + & x_2 & + & 2x_3 & + & x_4 & \geq & 3 \\
& x_1 & + & 2x_2 & + & x_3 & + & 2x_4 & \geq & 1 \\
& & & & & & & x_1, x_2, x_3, x_4 & \geq & 0.
\end{array}
$$

1) Write the dual of this linear program.

2) Is the solution $y_1^* = 0, y_2^* = \frac{1}{3}, y_3^* = 0, y_4^* = \frac{10}{3}, y_5^* = 0$. optimal for the dual? Use the method presented during the course and explain every step of the resolution.

Exercise 5 (3 points*, 20 minutes) **Difficult exercise, do not spend too much time on it.*
We consider the following algorithm for the set cover problem:

Data: a universe U of n elements, a collection of subsets of U $\mathcal{S} = S_1, ..., S_k$, and a cost function $c : S \rightarrow Q^+$.
Result: A set cover of U of minimum cost.

Find an optimal solution to the LP-relaxation. ;
(Rounding) Pick all sets S for which $x_S > 0$ in this solution. ;

Question: Show that this algorithm achieves a factor of f, with f the frequency of the most frequent element of U.

Remark: *In the algorithm presented during the class, only sets with $x_S \geq 1/f$ were chosen in the solution.*

Hint: *Use the primal complementary slackness conditions.*

3 Problem: Scheduling with class

Exercise 6 (7 points, 60 minutes) We consider in this problem two variants of scheduling problems. First the MINIMUM MAKESPAN SCHEDULING in which a job has the same processing time on any machine and, second, SCHEDULING ON UNRELATED PARALLEL MACHINES in which a job may have a different processing time on different machines. For these two problems, we will propose and study approximation algorithms.

The first problem is defined as follow.

Problem 1 (MINIMUM MAKESPAN SCHEDULING)) *Given processing times for n jobs, p_1, p_2, ..., p_n and an integer m, find an assignment of the jobs to m identical machines so that the completion time, also called the* makespan*, is minimized.*

We give below a simple factor 2 algorithm for this problem.

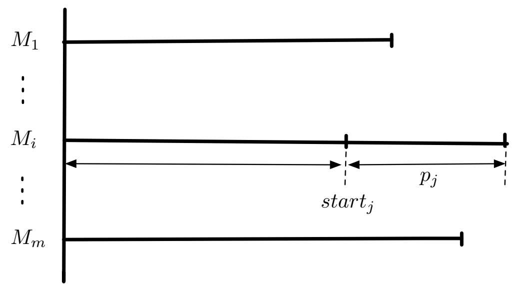
Algorithm 1 (Minimum Makespan Scheduling)

1. Order the jobs arbitrarily.
2. Schedule jobs on machines in this order, scheduling the next job on the machine that has been assigned the least amount of work so far.

Theorem: *The previous algorithm achieves an approximation guarantee of 2 for the minimum makespan problem.* We will prove this theorem.

1. **An example.** Consider $n = 5$ jobs of processing times $p_1 = 2, p_2 = 2, p_3 = 2, p_4 = 3, p_5 = 3$ and $m = 2$ machines.
   (a) Give the makespan given by the algorithm for the following order of jobs: job 2, job 1, job 4, job 3, job 5 (given at the step 1 of the algorithm).
   (b) Give the makespan given by the algorithm for the following order of jobs: job 4, job 5, job 1, job 2, job 3 (given at the step 1 of the algorithm).
2. **Two lower bounds.** The algorithm is based on two lower bounds on the optimal makespan, OPT:
   1. The largest processing time.
   2. $\frac{\sum_{i=1}^{n} p_i}{m}$.

   Explain the second lower bound.
3. We now prove the theorem. Consider the machine M_i that completes its jobs last in the schedule produced by the algorithm, and let j be the index of the last job scheduled on this machine. Let $start_j$ be the time at which job j starts execution on M_i.



   (a) Give an upper bound for $start_j$.

(b) Give an upper bound for p_j.

(c) Deduce a bound on the value of the scheduling returned by Algorithm 1.

4. **Tight example.** The factor 2 can be attained by some bad sequences of jobs. Provide an example of a sequence of $m^2 + 1$ jobs leading to a makespan of $2m$ for the proposed algorithm while the optimal schedule has a value OPT of $m + 1$.

5. **A better algorithm.** We have seen that the order of the jobs has a big importance on the result of the algorithm.

   Suggest another algorithm that could have an approximation ratio better than the one of Algorithm 1.

We know consider a second scheduling problem defined as follows

Problem 2 (Scheduling on unrelated parallel machines) *Given a set J of jobs, a set M of machines, and for each $j \in J$ and $i \in M$, $p_{ij} \in \mathbb{Z}^+$, the time taken to process job j by machine i, the problem is to schedule the jobs on the machine so as to minimize the* makespan*, i.e., the maximum processing time of any machine. We denote the number of jobs by n and the number of machine by m.*

Remark that, when for every job $j \in J$, we have $p_{ij} = p_{i'j}$ for all $i, i' \in M^2$, we go back to our first scheduling problem, where each job has the same running time on each machine.

6. Below is an integer linear program solving the SCHEDULING ON UNRELATED PARALLEL MACHINES. In this program, x_{ij} is a binary variable denoting whether job j is scheduled on machine i. There are two mistakes in this program:

$$\begin{array}{rll} \min & t & \\ \text{subject to} & & \\ & \sum_{i \in M} x_{ij} = 1 & (\forall i \in M) \\ & \sum_{j \in J} x_{ij} p_{ij} \geq t & (\forall i \in M) \\ & x_{ij} \in \{0, 1\} & (\forall i \in M, \forall j \in J) \end{array}$$

   Fix the two mistakes in the integer linear problem. Explain the program (constraints, objective function and what is t) and what were the mistakes.

7. We consider now the fractional relaxation of the corrected linear program.

   Example and integrality grap. Consider the example in which you have only one job of processing time m on any machine.

   (a) What is the minimum makespan, OPT?

   (b) What is the optimum value of the fractional relaxation?