Suggested solution to written exam
Introduction to linear and Integer programming (DM515)

Institut for Matematik & Datalogi
Syddansk Universitet

Monday, June 6, 2011, kl. 9–13

PROBLEM 1 Relaxation, the simplex method and the dual problem (20 %)

Consider the following integer programming problem (IP):

$$
\begin{array}{ll}
\text{maximize} & 3x_1 - x_2 + 2x_3 \\
\text{subject to} & x_1 - x_2 + x_3 \leq 5 \\
& 2x_2 + x_3 \leq 4 \\
& x_1 \leq 3 \\
& x_1, x_2, x_3 \geq 0,\ x_1, x_2, x_3 \in \mathbb{Z}
\end{array}
$$

Question a:

Write the LP relaxation (P1) of (IP) and explain why the objective value of an optimal solution to (P1) is an upper bound on the value of an optimal solution to (IP).

Answer a:

The LP relaxation is obtained by dropping the integrality constraint:

$$
\begin{array}{ll}
\text{maximize} & 3x_1 - x_2 + 2x_3 \\
\text{subject to} & x_1 - x_2 + x_3 \leq 5 \\
& 2x_2 + x_3 \leq 4 \\
& x_1 \leq 3 \\
& x_1, x_2, x_3 \geq 0
\end{array}
$$

As this increases the region of feasible solutions and we are dealing with a maximization problem, the value of an optimal solution to (P1) is an upper bound on the value of an optimal solution to (IP).

Question b:

Convert the problem (P1) to equational form by adding slack variables x_4, x_5, x_6 corresponding to the three in-equalities in the order from top to bottom. Next write the first simplex tableau with x_4, x_5, x_6 as the basic solution.

Answer b:

$$
\begin{array}{lrl}
\text{maximize} & 3x_1 - x_2 + 2x_3 & \\
\text{subject to} & x_1 - x_2 + x_3 + x_4 & = 5 \\
 & 2x_2 + x_3 + x_5 & = 4 \\
 & x_1 + x_6 & = 3 \\
 & x_1, x_2, x_3, x_4, x_5, x_6 & \geq 0
\end{array}
$$

The first simplex tableau with x_4, x_5, x_6 as basic solution is given by

$$
\begin{array}{rcl}
x_4 & = & 5 - x_1 + x_2 - x_3 \\
x_5 & = & 4 - 2x_2 - x_3 \\
x_6 & = & 3 - x_1 \\
\hline
z & = & 0 + 3x_1 - x_2 + 2x_3
\end{array}
$$

Question c:

Explain how x_1 may be brought into the basic solution and why this will increase the current objective value. Perform a pivot step that brings x_1 into the basis and explain how you select the variable to leave the basis in that step.

Answer c:

The variable x_1 has a positive coefficient in the last row. Hence increasing x_1 will increase the objective function. The maximum increase in x_1 is limited by the fact that all variables must remain non-negative. Hence it is the equation $x_6 = 3 - x_1$ that limits the growth of x_1 the most. We perform a pivot step for that row and get the following simplex tableau:

$$
\begin{aligned}
x_1 &= 3 - x_6 \\
x_4 &= 2 + x_2 - x_3 + x_6 \\
x_5 &= 4 - 2x_2 - x_3 \\
\hline
z &= 9 - x_2 + 2x_3 - 3x_6
\end{aligned}
$$

Question d:

After two more pivot steps (you do not have to perform these!) we obtain the following simplex tableau:

$$
\begin{aligned}
x_1 &= 3 - x_6 \\
x_2 &= \tfrac{2}{3} + \tfrac{1}{3}x_4 - \tfrac{1}{3}x_5 - \tfrac{1}{3}x_6 \\
x_3 &= \tfrac{8}{3} - \tfrac{2}{3}x_4 - \tfrac{1}{3}x_5 + \tfrac{2}{3}x_6 \\
\hline
z &= 13\tfrac{2}{3} - \tfrac{5}{3}x_4 - \tfrac{1}{3}x_5 - \tfrac{4}{3}x_6
\end{aligned}
$$

Argue that we have found an optimal solution to (P1). State the solution and its objective value.

Answer d:

All variables with a non-zero coefficient in the equation for z have a negative coefficient. This means that we cannot increase z by increasing any variable and as shown in the course this is equivalent to the fact that the current solution is optimal. The solution can be read out of the simplex tableau and it is

$$x_1 = 3, x_2 = \frac{2}{3}, x_3 = \frac{8}{3}$$

The objective value is $z = 13\frac{2}{3}$.

Question e:

Write up the dual problem (DP1) of (P1) where you use dual variables y_1, y_2, y_3 corresponding to the three in-equalities in (P1) from top to bottom.

Answer e:

The Dual problem is

$$\begin{array}{llr}
\text{minimize} & 5y_1 + 4y_2 + 3y_3 & \\
\text{subject to} & y_1 + y_3 & \geq 3 \\
& 2y_2 - y_1 & \geq -1 \\
& y_1 + y_2 & \geq 2 \\
& y_1, y_2, y_3 & \geq 0
\end{array}$$

Question f:

Show that $(y_1, y_2, y_3) = (\frac{5}{3}, \frac{1}{3}, \frac{4}{3})$ is an optimal solution to (DP1).

Answer f:

The vector $(y_1, y_2, y_3) = (\frac{5}{3}, \frac{1}{3}, \frac{4}{3})$ satisfies all three in-equalities (with equality) and has objective value $5 \cdot \frac{5}{3} + 4 \cdot \frac{1}{3} + 3 \cdot \frac{4}{3} = 13\frac{2}{3}$. This value is equal to the objective value of the primal solution that was found in (d) and hence is optimal by the weak duality theorem.

Question g:

As indicated in Question d, the optimal solution to (P1) is not an integer solution and hence not a solution to (IP). Use the last simplex tableau to derive the following Gomory cut (these cuts were introduced on weekly note 3):

$$\frac{2}{3}x_4 + \frac{1}{3}x_5 + \frac{1}{3}x_6 \geq \frac{2}{3} \tag{1}$$

Give a short explanation why this is a valid in-equality for (IP) (when we think of the slack variables being integer variables added to the original formulation), while adding (1) to (P1) will make the current optimal LP solution infeasible.

Answer g:

Rewriting the equation for x_2 in the last simplex tableau we get

$$x_2 - \frac{1}{3}x_4 + \frac{1}{3}x_5 + \frac{1}{3}x_6 = \frac{2}{3} \tag{2}$$

It was shown in one of the exercises on Weekly note 3 that this implies that the following in-equality (which is obtained by replacing each coefficient by the nearest smaller integer) is valid for all integer solutions to the problem (IP)

$$x_2 - x_4 \leq 0 \tag{3}$$

Now subtracting (3) from (2) we obtain (1).

PROBLEM 2 Flows(20 %)

At the annual party at a well-known academic institution on Fyn not all went smoothly and some participants had to be taken to medical emergency treatment at Odense University Hospital. In total 150 had to get a transfusion of one bag of blood. The hospital had 155 bags in stock. The distribution of blood groups in the supply and amongst the participants in need on blood is shown in the table below (the last part is writtem as demand).

Blood type	A	B	0	AB
Bags in stock	44	31	42	38
Demand	37	33	40	40

- Type A patients can only receive blood of type A or type 0.
- Type B patients can receive only type B or type 0.
- Type 0 patients can receive only type 0.
- Type AB patients can receive any of the four types.

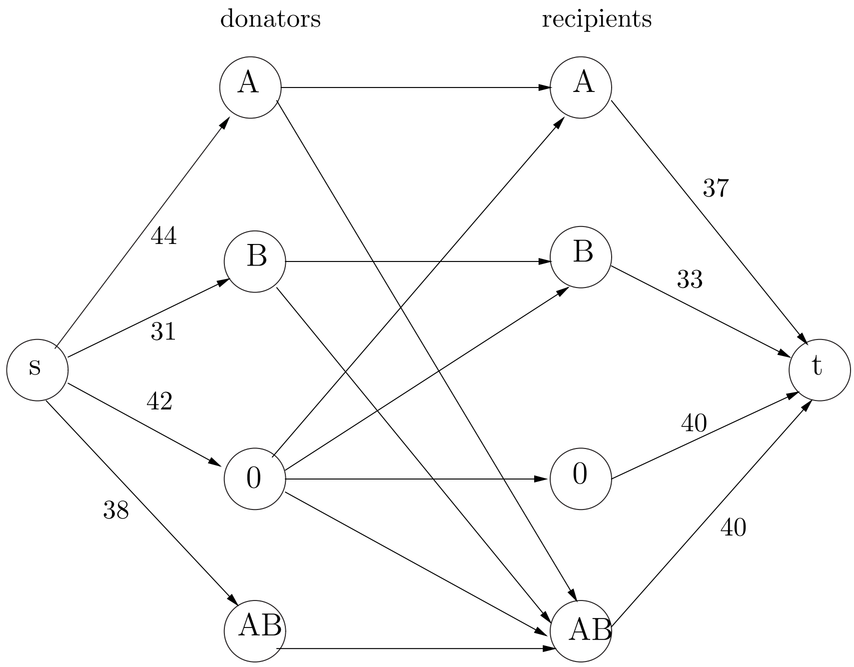


Figure 1: The proposed flow network $\mathcal{N}$ for the blood distribution problem. All capacities on the arcs in the middle are infinite.

Question a:

Consider the flow network $\mathcal{N}$ in Figure 1. Show that one can use this network to model the problem of checking whether it is possible to give all participants a blood transfusion that is compatible with the restrictions above. In particular, you should

- Explain the meaning of the arcs in the middle (what do they model?).
- Say what flow value ($b_x(s)$) you are looking for and how to interpret an (s,t)-flow of that value as a proper assignment of blood bags.
- You should also give an argument that we may replace all infinite capacities by finite numbers and say which (smallest) numbers will work.

Answer a:

- The capacities of the arcs from s indicate how much blood we have in stock of each type and the capacities on the arcs into t show the demand for each of the four types of recipients.
- The arcs in the middle model exactly which types of blood can be given to a patient with a given blood type.
- We want the flow to model the assignment of blood bags to participants so we are looking for an integer flow of value 150. The flow on an arc $\alpha \to \beta$ in the middle will then tell us how many bags of type α we will use to satisfy (a part of) the demand for participants with blood type β.
- We can replace the infinite capacities on arcs $\alpha \to \beta$ by the capacity of the arc from β to t (which equals the demand for that type).

Question b:

Show how to find a maximum (s,t)-flow in $\mathcal{N}$ by the augmenting path method (Ford-Fulkerson or shortest augmenting paths). You may start by listing four easy to find (s,t)-paths and send flow along these. After this you should show the resulting flow x and the residual network $\mathcal{N}(x)$ and show how to continue from there.

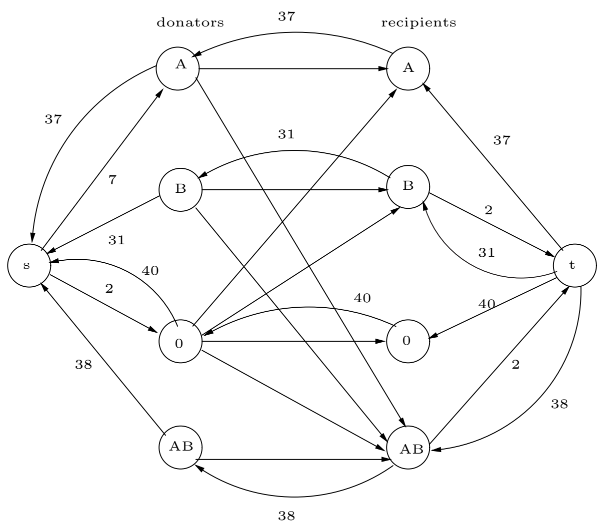
Answer b:

We start by sending flow along the following 4 paths which are disjoint apart from s,t:

- $sAAt$ 37 units
- $sBBt$ 31 units

- $s00t$ 40 units
- $sABABt$ 38 units

This gives us a flow of value 146. The residual network now looks as follows (infinite capacities not explicitly written)



Now we identify the following augmenting paths which again share only s, t: $s0Bt$ capacity 2 and $sAABt$ capacity 2. Augmenting along both of these gives us a maximum flow of values 150 (all arcs into t are filled).

Question c:

Show a feasible assignment of blood bags that is found via your flow algorithm above and say how you obtained it from your maximum (s, t)-flow.

.

Answer c:

The feasible assignment that can be read off from the final flow values are

- $A \rightarrow A$: 37 bags, $A \rightarrow AB$: 2 bags
- $B \rightarrow B$: 31 bags
- $0 \rightarrow 0$: 40 bags, $0 \rightarrow B$: 2 bags
- $AB \rightarrow AB$: 38 bags.

Question d:

Suppose now that five more participants show up also in need of a blood transfusion (so now all blood bags of the hospital would be needed if there is a solution). Use the network representation to analyse what their blood types must be in order for a feasible solution to exist.

Answer d:

Looking at the network model in Figure 1 we can see that participants who need blood of types B or 0 can only get this from blood types B or 0, so there is no room for extra demand here as we have 73=31+42 bags and currently need 33+40 = 73 bags. On the other hand participants with types A and AB can be supplied by blood type A of which we still have 5 bags in stock so we can handle the extra participants if and only if their blood types are all in the set $\{A, AB\}$.

PROBLEM 3 Project scheduling (15 %)

A small project has 6 sub-activities A, B, C, D, E, F whose individual dependency (shown by the immediate predecessors) is given in Figure 2. Here we also list the normal time (in weeks), the absolute minimum time and the cost of shortening the activity by one week.

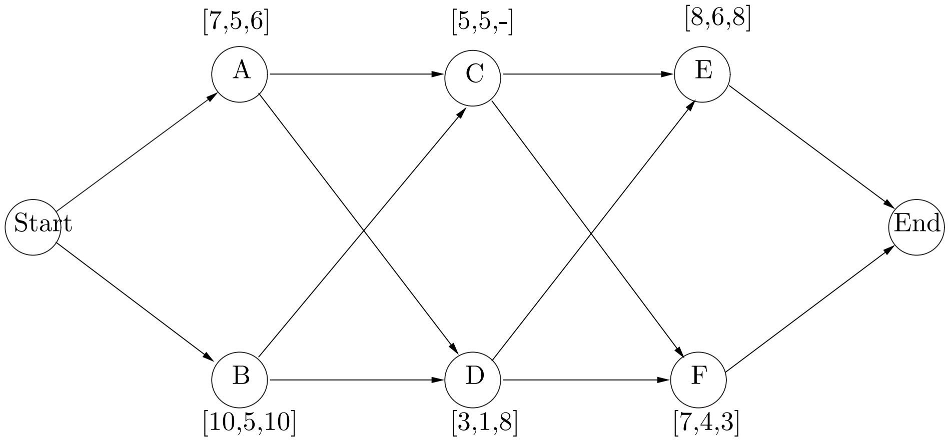


Figure 2: An AON network for a small project with 6 activities. For each activity the following data is given in that order from left to right: normal time, minimum time in weeks, and the cost of shortening the duration of the activity by one week.

Question a:

Describe (in words) an algorithm for finding the duration of a given project when it is modelled as an AON network and all activities are at the normal duration. What is the running time of the algorithm?

Answer a:

As we have seen in the course we can calculate the duration of the whole project by calculation the earliest finishing times EF(i) for each activity. This can be done by finding an acyclic ordering of the project graph in linear time (via DFS) and then calculating EF(i) via the formula $EF(i) = \max_{j \to i}\{EF(j) + d_i\}$

Question b:

Illustrate your algorithm on the project network in Figure 2 and state the duration of the project found by the algorithm. It suffices to show a few steps and then write the solution. You must also show the result of the same calculation when all activities are at the minimum duration. Here it is enough to show the final result (you may also show both calculations on one graph).

Answer b:

Both results are shown in Figure 3 (the calculation is not shown).

The duration when all activities are at their normal duration is 23 weeks and it is 16 weeks if all activities are at their minimum duration.

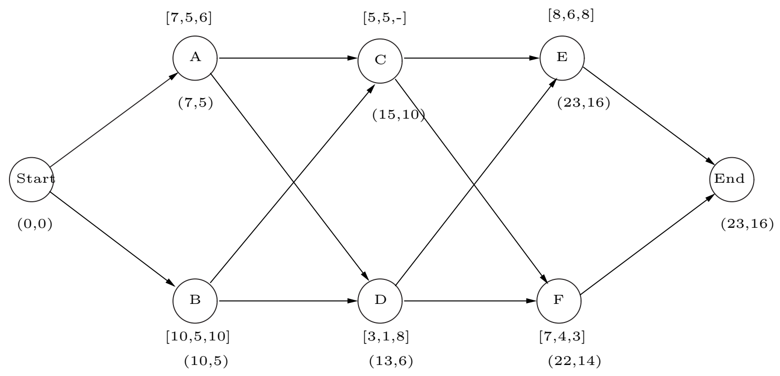


Figure 3:

Question c:

The goal is now to shorten the duration of the project to 19 weeks. This means that the duration of one or more activities has to be shortened. Of course we want to select these so that the total cost of shortening the duration to 19 weeks is minimized. Formulate this problem a linear programming problem and argue that the optimal solution to this LP will provide the correct answer. Note that you must use the actual data in the LP formulation!

Answer c:

We will use a variable x_i to indicate how much we will shorten activity i and another set of variables y_i which will indicate the earliest starting time of activity i. For each arc $i \to j$ in the project network we will add the constraint $y_j \geq y_i + (d_i - x_i)$. We also use a variable y_{end} to express that the dummy activity "end" cannot start before all its immediate predecessors have finished. Finally we add the constraint $y_{end} \leq 19$ to force the total project time to be lowered to 19.

$$
\begin{aligned}
\text{minimize} \quad & 6x_A + 10x_B + 8x_D + 8x_E + 3X_F \\
\text{subject to } y_C \quad & \geq y_A + (7 - x_A) \\
y_C \quad & \geq y_B + (10 - x_B) \\
y_D \quad & \geq y_A + (7 - x_A) \\
y_D \quad & \geq y_B + (10 - x_B) \\
y_E \quad & \geq y_C + 5 \\
y_E \quad & \geq y_D + (3 - x_D) \\
y_F \quad & \geq y_C + 5 \\
y_F \quad & \geq y_D + (3 - x_D) \\
y_{end} \quad & \geq y_E + (8 - x_E) \\
y_{end} \quad & \geq y_F + (7 - x_F) \\
y_{end} \quad & \leq 19 \\
x_A \quad & \leq 2 \\
x_B \quad & \leq 5 \\
x_D \quad & \leq 2 \\
x_E \quad & \leq 2 \\
x_F \quad & \leq 3 \\
x_A - x_F \quad & \geq 0,\ y_A, y_B, y_C, y_D, y_E, y_F, y_{end} \geq 0
\end{aligned}
$$

The optimal solution to this LP will tell us to shorten activity i by $x_i \geq 0$ units and since the cost we apply to each x_i is the per unit shortening cost of that activity, the cost of the solution will be that of shortening the project in the way suggested by the x_i's. Conversely, any feasible shortening of projects corresponds to a solution to this LP whose cost (in the LP) is the actual cost of shortening the activities in the way suggested.

PROBLEM 4 Branch and Bound (15 %)

Question a:

Consider the following knapsack problem:

$$\begin{aligned} \max \quad & 50x_1 + 60x_2 + 140x_3 + 40x_4 \\ \text{such that} \quad & 5x_1 + 10x_2 + 20x_3 + 20x_4 \leq 30 \\ & x_1, x_2, x_3, x_4 \in \{0, 1\} \end{aligned}$$

State the LP-relaxation and show how to find an optimal solution to this without using the simplex method. Hint: a greedy approach works.

Answer a:

We obtain the LP relaxation by replacing the integrality constraints on the x_i's by $0 \leq x_i \leq 1$, $i = 1, 2, 3, 4$. It is well-known that this (so-called) fractional knapsack problem can be solved to optimality by the greedy algorithm. First order the items in decreasing order according to the values of $r_i = \frac{v_i}{w_i}$, where v_i is the value and w_i the weight/size of item i. In our case $(r_1, r_2, r_3, r_4) = (\frac{50}{5}, \frac{60}{10}, \frac{140}{20}, \frac{40}{20}) = (10, 6, 7, 2)$ so this gives $r_1 > r_3 > r_2 > r_4$. Now consider the items in the order given by the ordering above and take the maximum amount possible of the current item. This will give the optimum solution since we pack at maximum relative value per unit. The solution found in this way is $x_1 = x_3 = 1$ and $x_2 = \frac{1}{2}$ with value 220.

Question b:

Solve the knapsack problem above to optimality using branch and bound. Use the depth first search strategy when exploring the nodes of the B&B tree and when you branch, you should first explore the branch corresponding to including the item that you are branching on. Show the B&B tree in each step and explain briefly (with justification) what you conclude in each step of the algorithm. This includes which nodes in the B&B tree you must continue exploring and which nodes you can finish (and with what conclusion).

Answer b:

We use the value of the fractional knapsack problem as the upper bound function.

First we solve this as above to get the upper bound 220 for the maximum objective value. Since x_2 is not an integer we branch on x_2 and get two branches, one where we set $x_2 = 1$ and one where $x_2 = 0$.

As suggested we first treat the $x_2 = 1$ branch. The remaining capacity is 20 so the optimal solution to the resulting fractional knapsack problem

$$\begin{aligned} \max \quad & 50x_1 + 60x_2 + 140x_3 + 40x_4 \\ \text{such that} \quad & 5x_1 + 10x_2 + 20x_3 + 20x_4 \le 20 \\ & 0 \le x_1, x_2, x_3, x_4 \le 1 \end{aligned}$$

is $x_1 = 1$ and $x_3 = \frac{3}{4}$ with objective value 215. As x_3 is fractional we branch on that and first consider the branch $x_3 = 1$ (and $x_2 = 1$). This problem is trivial to handle since the knapsack is full when we take items 2 and 3 and the value of the integer solution is 200. Now we have a feasible solution which we can use in the bounding process from now on. Now consider the $x_3 = 0$ branch (that is we currently have $x_2 = 1$ and $x_3 = 0$). Here the optimal solution to the LP relaxation is $x_1 = 1$ and $x_4 = \frac{3}{4}$ with value 140. Since 140 is less than 200, the value our known feasible solution we do not continue branching but backtrack instead, hence completing the $x_2 = 1$ branch.

Now consider the branch for $x_2 = 0$. The LP optimum is now $x_1 = x_3 = 1$ and $x_4 = \frac{1}{4}$ with value 200. This is no better than the value of of known integer solution so we stop that branch and hence the whole search.

The optimum solution is hence $x_2 = x_3 = 1$ with value 200.

PROBLEM 5 Formulation of IP problems and the cutting plane method (20 %)

Let $G = (V, E)$ be a connected graph on n vertices $V = \{1, 2, \ldots, n\}$ and non-negative weights w_e on its edges. The fact that we can solve the minimum spanning tree problem in polynomial time is ignored in this problem.

Question a:

Explain why the following is a correct integer programming formulation of the minimum spanning tree problem and state the LP-relaxation:

$$\begin{aligned} \text{Minimize} \quad & \textstyle\sum_{e \in E} w_e x_e && (4) \\ \text{such that} \quad & \textstyle\sum_{e \in E} x_e = n - 1 && (5) \\ & \textstyle\sum_{\{e : |e \cap S| = 1\}} x_e \geq 1 \ \forall \ \emptyset \neq S \subseteq V - \{1\} && (6) \\ & x_e \in \{0, 1\} && (7) \end{aligned}$$

For later reference we call this formulation MSTIP and its LP-relaxation MSTLP.

Answer a:

The conditions (5), (6) and (7) imply that the edges with $x_e = 1$ form a spanning tree and since we minimize the cost of these edges, the claim follows. The LP relaxation is obtained by replacing (7) by $0 \leq x_e \leq 1 \ \forall e \in E$.

The conditions (6) are not nice to work with, since there are exponentially many of those, so even MSTLP may be impossible to solve when n gets large.

Consider the following integer programming problem MST-NO-CUTS:

$$\begin{aligned} \text{Minimize} \quad & \textstyle\sum_{e \in E} w_e x_e && (8) \\ \text{such that} \quad & \textstyle\sum_{e \in E} x_e = n - 1 && (9) \\ & x_e \in \{0, 1\} && (10) \end{aligned}$$

Question b:

Explain how to solve the LP-relaxation of MST-NO-CUTS. Why does this linear programming problem always have an optimal integer solution x^*?

Answer b:

First observe that if e, e' are two edges such that $w_e < w_{e'}$ then we must have $x_e \geq x_{e'}$ in any optimal solution to the LP relaxation of MST-NO-CUTS, since otherwise we could get a cheaper solution be lowering $x_{e'}$ and increasing x_e. Secondly, if e, e' are two edges such that $w_e = w_{e'}$, then we can make a solution of the same cost by lowering $x_{e'}$ and increasing x_e (if that is possible). These two observations imply that it is optimal to select the $n-1$ edges of lowest cost: If there are more than $n-1$ edges with $x^*_e > 0$ then we could get a new optimal solution by increasing the value on the cheapest edge that currently has $0 < x^* < 1$.

Question c:

What can you say about the objective value of the optimum solution x^* to MST-NO-CUTS compared to the optimal solution to MSTIP (the minimum spanning tree solution)?

Answer c:

The objective value of x^* is a lower bound for the optimum objective value for MSTIP because MST-NO-CUTS contains only part of the constraints of MSTIP (we have dropped (6))

Question d:

How do you check (algorithmically and in polynomial time!) whether x^* above is also a feasible solution to MSTIP? What can you conclude if it is a feasible solution to MSTIP?

Answer d:

Since x^* is integer valued, all we have to check is that the graph G consisting of those edges with $x^*_e = 1$ is connected, since then it will be a spanning tree. This can be done by applying DFS to G from an arbitrary vertex, say vertex 1.

Question e:

Suppose now that x^* is a not feasible solution to MSTLP, that is, the condition (6) is violated for some X. Explain how to find at least one cut $(X, V - X)$ with $\emptyset \neq X \subseteq V - \{1\}$ such that adding the condition $\sum_{\{e:|e \cap X|=1\}} x_e \geq 1$ to the LP relaxation of the MST-NO-CUTS formulation makes x^* infeasible for this extended problem.

Answer e:

As we saw above in the answer to d: x^* is feasible for MSTIP if and only if x^* is the incidence vector of a spanning tree. If this is not the case, then the DFS that we apply above will not reach all vertices but only a proper subset X of the vertices. Now we can add the condition $\sum_{\{e:|e\cap X|=1\}} x_e \geq 1$ and make x^* infeasible.

Suppose that we have iterated this process a number of times and that y^* is an optimum solution to the current LP problem (obtained from the LP-relaxation of MST-NO-CUTS by adding the violated cuts found so far).

Question f:

Explain briefly how to use a maximum flow algorithm (as a subroutine) to check whether y^* satisfies (6).

Answer f:

Define a flow network $\mathcal{N} = (V, A, \ell \equiv 0, u)$ where $V = \{1, 2, \ldots, n\}$ and $A = \{ij | y^*_e > 0 \text{ and } e \text{ has ends } i, j\}$. We also take $u_{ij} = y^*_e$ for each such arc (and its opposite). Now the condition (6) holds if and only if there is no cut in $\mathcal{N}$ of capacity less that 1. By the max flow min cut theorem, this can be checked via $n-1$ maxflow calculations by checking whether $\mathcal{N}$ has a $(1, i)$-flow of value at least 1 for all $i \in \{2, 3, \ldots, n\}$.

PROBLEM 6 Maximum weight matchings in bipartite graphs (10 %)

Consider the maximum weight matching problem in a bipartite graph $G = (V, E)$ with a non-negative weight function w on the edges, as described on page 33 in MG:

$$\begin{array}{rc} \text{Maximize} & \sum_{e \in E} w_e x_e \\ \text{such that} & \sum_{\{e \in E: v \in e\}} x_e = 1 \ \forall v \in V \\ & x_e \in \{0, 1\} \ \forall e \in E \end{array} \tag{11}$$

Suppose that we are given an optimal solution x^* to the LP-relaxation of (11) for a bipartite graph G on 6 vertices such that for some 6-cycle $abcdefa$ of G x^* has the values $x^*_{ab} = x^*_{cd} = x^*_{ef} = \frac{1}{3}$ and $x^*_{bc} = x^*_{de} = x^*_{af} = \frac{2}{3}$ (see Figure 4).

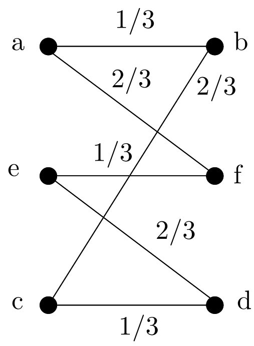


Figure 4:

Question a:

Prove (without knowing what the cost of the edges are!) that the edges af, de, bc form a maximum weight matching in G. Hint: Consider the proof of Theorem 3.2.1 in MG.

Question b:

Can you give a different maximum weight matching?

Answer to a and b:

Let x^* be as given in the figure and the the unknown costs be $c_{ab}, c_{af}, c_{bc}, c_{cd}, c_{ef}, c_{de}$. Let z^* be the cost of x^*. If we lower the values of x^* by some $\epsilon \leq \frac{1}{3}$ on the edges ab, cd, ef and increase it by ϵ on the edges af, de, bc then we obtain a new feasible solution of cost

$z_1 = z^* + \epsilon[(c_{af} + c_{bc} + c_{de}) - (c_{ab} + c_{cd} + c_{ef})]$. On the other hand if we lower the values of x^* by the same ϵ on the edges af, de, bc and increase by ϵ on the edges ab, cd, ef, then we get a new feasible solution of cost $z_2 = z^* + \epsilon[(c_{ab} + c_{cd} + c_{ef}) - (c_{af} + c_{bc} + c_{de})]$. As x^* is optimal and has weight z^* both $z_1 \leq z^*$ and $z_2 \leq z^*$ holds implying that $[(c_{ab} + c_{cd} + c_{ef}) - (c_{af} + c_{bc} + c_{de})] = 0$ and both of our new solutions are optimal.

Thus taking $\epsilon = \frac{1}{3}$ we get from the optimality of z_1 that $x_{ab} = x_{cd} = x_{ef} = 0$ and $x_{bc} = x_{de} = x_{af} = 1$ is an optimal integer valued solution to the LP relaxation and hence corresponds to the maximum weight matching bc, de, af in G.

Similarly the optimality of z_2 and $\epsilon = \frac{1}{3}$ implies that ab, cd, ef is also a maximum weight matching of G.