

Written Exam

Introduction to Linear and Integer Programming (DM515)

Department of Mathematics and Computer Science
University of Southern Denmark

Friday, June 15, 2012, 10:00–14:00, U49 and U49D

The exam consists of answers to a number of tasks divided into subtasks, which are to be handed in electronically in <http://e-learn.sdu.dk> (Blackboard).

- The content of the document “Exam Instructions” published in the public course Web Page is assumed known.
- *Remember to justify all your statements!* You may refer to results from the books or the lecture slides listed at the course web page. In particular, it is possible to justify a statement by saying that it derives trivially from a result in the textbook (if this is true). You may use all methods or extensions that have been used in the assignment sheets, published during the course. However, it is not allowed to answer a subtask exclusively by reference to an exercise seen during the course. Reference to other books (outside the course material) or to internet links is not accepted as answer to a task.
- The weight in the evaluation of each task is given in percent. The weight is redistributed on the subtasks. Higher weights are set on tasks and subtasks that constitute necessary minimal knowledge to pass the exam. Weights are not representative of the difficulty of the task or the amount of effort needed to solve it.
- The exam consists of 6 tasks distributed on 24 pages.
- You may write your answers in Danish or in English.

Task 1 (25%)

A small firm that produces two types of mattresses, Single and Double, has two workers for the final treatment of the mattresses before being ready for sell. In one hour the two workers can process the following number of mattresses:

Worker A: 3 Single mattresses and 1 Double mattress;

Worker B: 2 Single mattresses and 2 Double mattresses

The market demands at least 30 Single and 20 Double mattresses per day. The cost of one hour of working time is 90 Dkk for worker A and 80 Dkk for worker B. The management of the firm wants to determine the number of hours per day that each worker has to be engaged in order to meet the demand at the minimum cost. Workers can only be engaged for multiples of hours and not for fractions of hours.

Subtask 1.a

Formulate the problem as an integer linear programming problem. (Ignore that workers can have a maximum number of working hours per day.)

Solution

$$\begin{aligned} z = \text{minimize} \quad & 90 x_A + 80 x_B \\ \text{subject to} \quad & 3 x_A + 2 x_B \geq 30 \\ & x_A + 2 x_B \geq 20 \\ & x_A \geq 0 \text{ and integer} \\ & x_B \geq 0 \text{ and integer} \end{aligned}$$

In GLPK:

```
var x_1 integer >= 0;
var x_2 integer >= 0;
var x_3 integer >= 0;

minimize z: 90 * x_A + 80 * x_B ;

subject to
S:      3 *x_A + 2 *x_B      >= 30;
D:      x_A + 2* x_B      >= 20;
solve;
end;
```

Subtask 1.b

Write the linear programming relaxation (P1) of the original problem formulation (P) given at the previous subtask.

Explain the relation between the optimal solution of (P1) and the optimal solution of (P).

Solution

$$\begin{aligned} z_{P1} = \text{minimize} \quad & 90 x_A + 80 x_B \\ \text{subject to} \quad & 3 x_A + 2 x_B \geq 30 \\ & x_A + 2 x_B \geq 20 \\ & x_A \geq 0 \\ & x_B \geq 0 \end{aligned}$$

z_{LP} is a lower bound of z .

Subtask 1.c

Derive the dual (D1) of (P1) using the Lagrange multipliers approach. Show all steps of the derivation and write the final problem (D1).

Solution We introduce the multipliers $y_1, y_2 \leq 0$ and relax the two constraints in the objective function:

$$P(y_1, y_2) = \min 90x_A + 80x_B + y_1(3x_A + 2x_B - 30) + y_2(x_A + 2x_B - 20)$$

Thus

$$\forall y_1, y_2 \leq 0 \quad P(y_1, y_2) \leq z$$

Hence $P(y_1, y_2)$ is a lower bound for P1. The best lower bound will be given by

$$\begin{aligned} z_{D1} &= \max_{y_1, y_2 \leq 0} \left\{ \min_{x_A, x_B \geq 0} 90x_A + 80x_B + y_1(3x_A + 2x_B - 30) + y_2(x_A + 2x_B - 20) \right\} \\ &= \max_{y_1, y_2 \leq 0} \left\{ \min_{x_A, x_B \geq 0} (90 + 3y_1 + y_2)x_A + (80 + 2y_1 + 2y_2)x_B - 30y_1 - 20y_2 \right\} \end{aligned}$$

The optimization problem

$$\min_{x \geq 0} c_1x_1 + c_2x_2 + \dots + c_nx_n$$

can be solved by inspection as follows:

$$\begin{aligned} \text{if } c_i < 0 &\Rightarrow x_i = \infty \\ \text{if } c_i \geq 0 &\Rightarrow x_i = 0 \end{aligned}$$

Hence the best lower bound will be:

$$\begin{aligned} z_{D1} &= \max -30y_1 - 20y_2 \\ \text{st } &(90 + 3y_1 + y_2) \geq 0 \\ &(80 + 2y_1 + 2y_2) \geq 0 \\ &y_1, y_2 \leq 0 \end{aligned}$$

or equivalently

$$\begin{aligned}
 z_{D1} = \max & 30y_1 + 20y_2 \\
 \text{st } & 3y_1 + y_2 \leq 90 \\
 & 2y_1 + 2y_2 \leq 80 \\
 & y_1, y_2 \geq 0
 \end{aligned}$$

Subtask 1.d

- Put (P1) in standard form and write the corresponding tableau.
- Bring the tableau in canonical form.
- Indicate the basic solution of the canonical form.
- State whether the solution is feasible.

Solution After introduction of surplus variables by means of which we transform the two constraints in equalities preserving positive right hand side terms we obtain the first tableau.

Tableau #1

x_A	x_B	s1	s2	-z	b
3	2	-1	0	0	30
1	2	0	-1	0	20
90	80	0	0	1	0

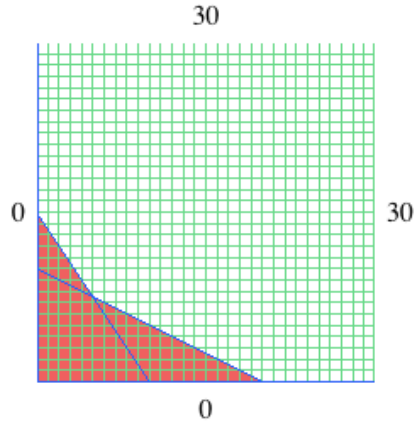
The canonical form has the identity matrix as a submatrix of the constraint matrix. We obtain it from the first tableau by multiplying by -1 the first and second row.

Tableau #2

x_A	x_B	s1	s2	-z	b
-3	-2	1	0	0	-30
-1	-2	0	1	0	-20
90	80	0	0	1	0

Since the RHS terms are negative the basic solution corresponding to this tableau is infeasible (surplus and slack variables are always defined to be non-negative).

Figure 1 shows the feasibility region graphically.



The feasible region is shown in white.

Figure 1:

Subtask 1.e

After two iterations of the appropriate form of the simplex method we obtain the two following tableaux:

Tableau #4

	x_A	x_B	s1	s2	-z	b
I	1	2/3	-1/3	0	0	10
II	0	-4/3	-1/3	1	0	-10
III	0	20	30	0	1	-900

Tableau #5

	x_A	x_B	s1	s2	-z	b
I	1	0	-1/2	1/2	0	5
II	0	1	1/4	-3/4	0	15/2
III	0	0	25	15	1	-1050

Explain in detail which operations have been done to reach the two tableaux. In particular, indicate the pivot of the two iterations, the entering and leaving variables and the elementary row operations to go from one tableau to the other. (Use the names given in the first column of the tableaux to identify rows.)

Solution We applied two iterations of the dual simplex. From tableau # 2 to # 3:

1. the pivot must be < 0
2. select a row i^* such that $b_{i^*} < 0 \rightarrow$ the first row (-30)
3. select the column $j^* = \arg \min_j |c_j/a_{i^*j}| \Rightarrow \min\{90/3 = 30, 80/2 = 40\} \Rightarrow j^* = 1$

Hence the pivot is -3 , x_A enters the basis and $s1$ leaves it. The elementary row operations are:

$$\begin{aligned}
 I' &= (-1/3)I \\
 II' &= II + I' \\
 III' &= III - 90I'
 \end{aligned}$$

From tableau # 3 to # 4:

1. the pivot must be < 0
2. select a row i^* such that $b_{i^*} < 0 \rightarrow$ the second row (-10)
3. select the column $j^* = \arg \min_j |c_{j^*}/a_{i^*j}| \Rightarrow \min\{20*3/4 = 15, 30*3 = 90\} \Rightarrow j^* = 2$

Hence the pivot is $-4/3$, x_B enters the basis and s_2 leaves it. The elementary row operations are:

$$\begin{aligned} I' &= I - 2/3II' \\ II' &= (-3/4)II \\ III' &= III - 20I' \end{aligned}$$

Subtask 1.f

Determine the optimal solution of (P1) and the value of the corresponding objective function. (Report the values for all four variables of the standard form in the optimal solution and remember to explain why the solution you are reporting is feasible and optimal.)

Solution We can read the optimal solution from Tableau # 4. It is $[5, 15/2, 0, 0]$ and it has value 1050. It is feasible because all RHS terms are positive and optimal because all reduced costs are positive.

Subtask 1.g

Determine the value of the dual variables.

Solution From the strong duality theorem we know that the dual variables get the value of the reduced cost of the slack variables with a change of sign. Hence, $y_1 = -25$ and $y_2 = -15$.

Subtask 1.h

The optimal integer solution is found by adding a Gomory cut.

- Derive the cut.
- Write it in the space of the original variables.
- Argue shortly that the cut is a valid inequality for (P) and that it will make the current optimal solution of (P1) infeasible.

Solution During the course we saw that the Gomory cut can be derived as follows:

$$\sum_{j \in N} (\bar{a}_{uj} - \lfloor \bar{a}_{uj} \rfloor) x_j \geq \bar{b}_u - \lfloor \bar{b}_u \rfloor$$

We focus on the second row of the final tableau since that is the only one with a fractional value in the RHS term.

$$\begin{aligned} (1/4 - 0)s_1 + (-3/4 - (-1))s_2 &\geq (15/2 - 7) \\ 1/4s_1 + 1/4s_2 &\geq 1/2 \\ s_1 + s_2 &\geq 2 \end{aligned}$$

To express the constraint in the space of the original variables we need to substitute s_1 and s_2 as given from the first tableau:

$$\begin{aligned} s_1 &= 3x_A + 2x_B - 30 \\ s_2 &= x_A + 2x_B - 20 \end{aligned}$$

Hence:

$$-30 + 3x_A + 2x_B - 20 + x_A + 2x_B \geq 2 \tag{1}$$

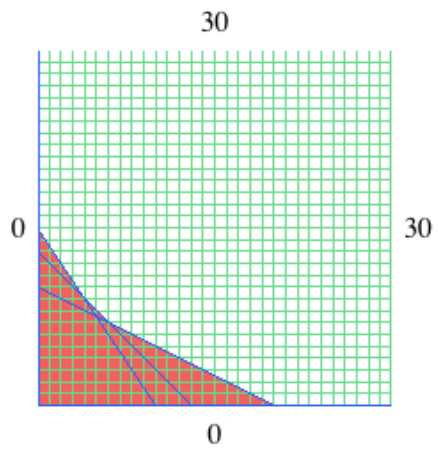
$$-50 + 4x_A + 4x_B \geq 2 \tag{2}$$

$$4x_A + 4x_B \geq 52 \tag{3}$$

$$x_A + x_B \geq 13 \tag{4}$$

$$\tag{5}$$

For the current solution $x = (5, 15/2, 0, 0)$ the inequality is not satisfied: $25/2 \not\geq 13$, hence the cut is valid and it tightens the formulation. We can see this also graphically in Figure 2.



The feasible region is shown in white.

Figure 2:

Task 2 (20%)

Consider the network $\mathcal{N} = (V, A, l \equiv 0, u, b, c)$ of Figure 3 and let x^* be the flow there represented.

Subtask 2.a

Argue that the flow x^* is feasible and give its cost.

Solution The flow is feasible because:

1. it satisfies all capacity bounds constraints
2. it satisfies all demands at the nodes

Its cost is 30.

Subtask 2.b

Draw the residual network $\mathcal{N}(x^*)$ explaining briefly how you find the capacity on the arcs of $\mathcal{N}(x^*)$ (it suffices to show a couple of examples).

Solution

Example:

$$\begin{aligned} r_{fd} &= (u_{fd} - x_{fd}) + (x_{df} - l_{df}) = 3 - 3 + 0 - 0 = 0 \\ r_{df} &= (u_{df} - x_{df}) + (x_{fd} - l_{fd}) = 0 - 0 + 3 - 0 = 0 \end{aligned}$$

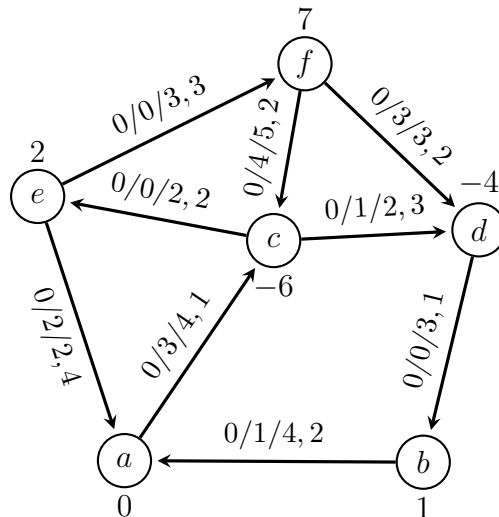
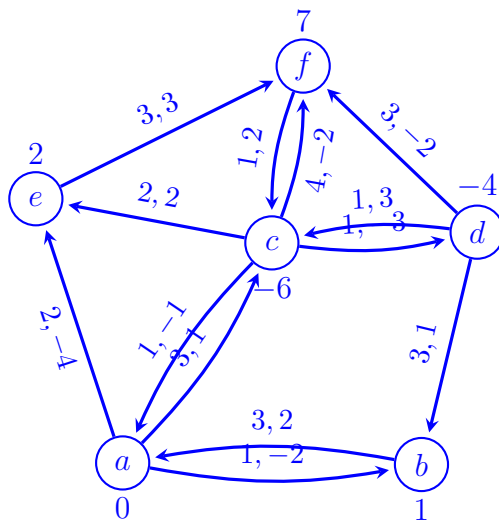


Figure 3: A network with lower bound, flow amount, upper bound and cost on arcs in the format $l_{ij}/x_{ij}^*/u_{ij}, c_{ij}$ and demands b_i on the vertices. [The code of this figure is available in the online version of this document.]



Subtask 2.c

Argue that x^* is a minimum cost cycle for \mathcal{N} .

Solution There is no negative cost cycle in the residual network, hence for the min cost optimality theorem the flow is optimal.

Subtask 2.d

Suppose now that the lower bound to the flow in the arc ef is set to 1 while all the rest remains unchanged, thus yielding the network $\mathcal{N}' = (V, A', u, b, c)$. Find a new feasible flow and explain how you found it.

Solution To find a feasible flow we need to transform the network removing lower bounds on arcs. This is done as seen in class by decreasing the balance value of tail node by 1 and increasing the one of the head node by the same amount. We then construct an st network as shown in Figure 4.

This is an instance of max flow that we solve by augmenting path algorithm (Ford-Fulkerson). We find:

```

sfdt 3
sfcdt 1
sfct 4
seact 1
sbact 1

```

The flow saturates the source arcs hence it is a feasible flow.

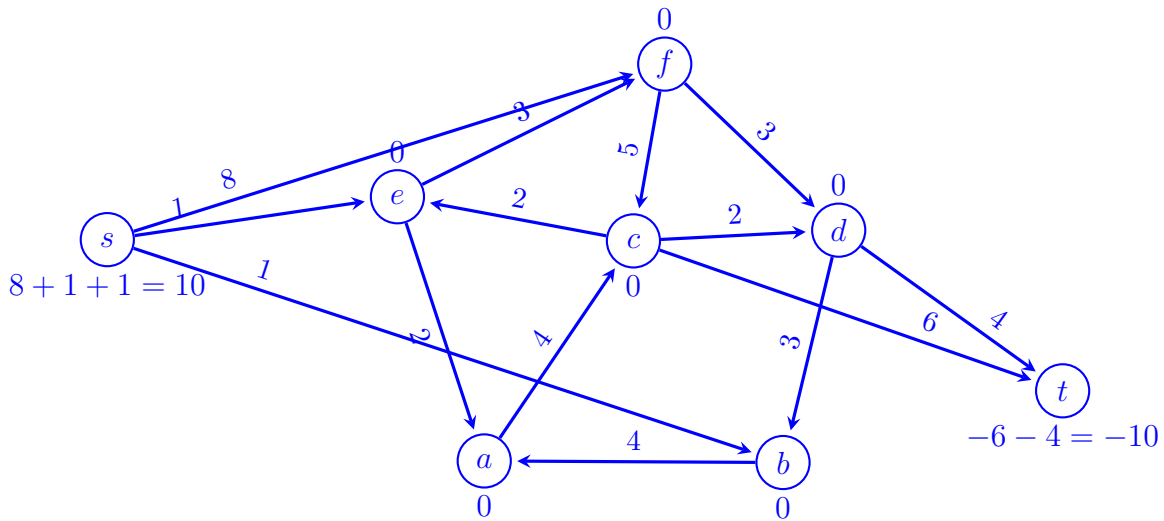


Figure 4: .

Subtask 2.e

Find an optimal cost flow x' for \mathcal{N}' .

Solution From the previous task the feasible flow is represented in Figure 5. We obtain it going backwards in the transformations applied earlier. (In specific, we reintroduced the lower bound and hence established $x_{ef} = 1$.) In the same figure right the residual network in which there are no negative cost cycles.

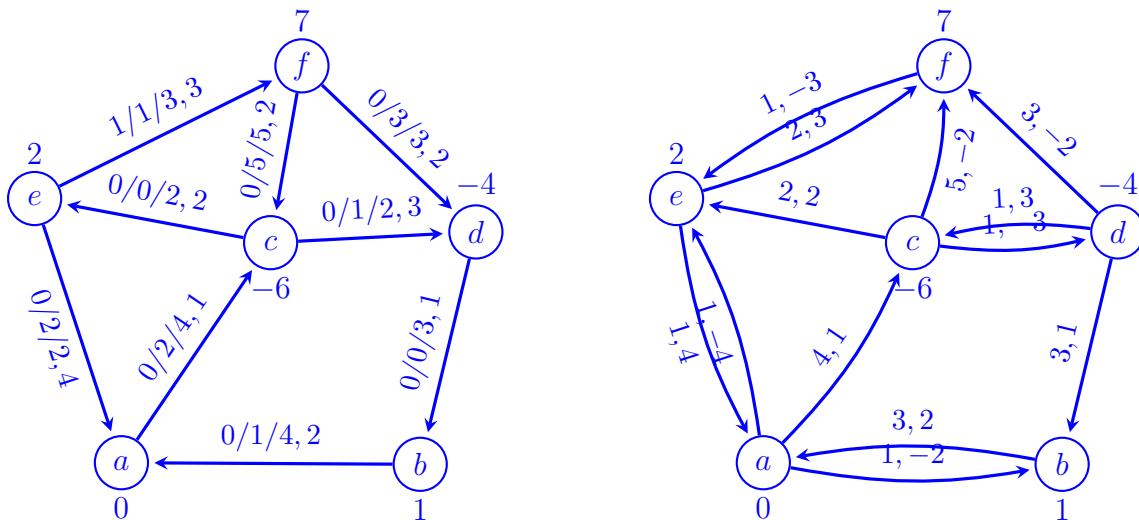


Figure 5:

It occurs that the new flow is obtained by augmenting the previous one through a zero

cost cycle $fcaef$ in the residual network $\mathcal{N}(x^*)$. Hence the cost of the flow remains the same and there continue not to be improving cycles, hence it is optimal.

Subtask 2.f

Decompose x' into path and cycle flows and explain how you found each single component in the decomposition.

Solution We apply DFS from any vertex with $b(v) > 0$ and finish the path in a vertex with $b(v) < 0$ or in a cycle. The decomposition yields only paths:

```
fc 5
fd 3
ba 1
ac 1
eacd 1
ef 1
```

Task 3 (15%)

The Danish Research Council has to decide which research projects to finance. The total budget for the projects is 20 million Dkk. The table below shows the evaluation from 0 (worst) to 2 (best) that the projects received by the external reviewers and the amount of money required.

	1	2	3	4	5
Evaluation score	1	1.8	1.4	0.6	1.4
Investment (in million of DKK)	6	12	10	4	8

Projects 2 and 3 have the same coordinator and the Council decided to grant only one of the two.

The Council wants to select the combination of projects that will maximize the total relevance of the projects, that is, the sum of the evaluation score while remaining within the budget.

Subtask 3.a

Formulate the problem of deciding on which project the Council has to invest as an integer linear programming problem P .

Solution In .lp format:

```
\* Problem: lp3 *\n\nMaximize\n  tot: + x(1) + 1.8 x(2) + 1.4 x(3) + 0.6 x(4) + 1.4 x(5)\n\nSubject To\n  budget: + 6 x(1) + 12 x(2) + 10 x(3) + 4 x(4) + 8 x(5) <= 20\n  a: + x(2) + x(3) <= 1\n\nBounds\n  0 <= x(1) <= 1\n  0 <= x(2) <= 1\n  0 <= x(3) <= 1\n  0 <= x(4) <= 1\n  0 <= x(5) <= 1\n\nEnd
```

Subtask 3.b

We want the IP instance solved using the branch-and-bound algorithm. What is the optimal solution x^* to the LP relaxation P' ? [Hint: use `glpsol` to find out. You find a skeleton with the input data in the online version of this document.]

Solution The following glpk model gives solution $\vec{x} = [1, 0.5, 0, 0, 1]$ and $z = 3.3$.

```
set P;
param score{P};
param crone{P};
param B;

var x{i in P} >=0, <=1;

maximize tot: sum{i in P} score[i]*x[i];

subject to
  budget: sum{i in P} crone[i]*x[i] <= B;
  a: x[2]+x[3]<=1;
# b: x[2]<=0;
# c: x[5]<=0;
# c: x[3]>=1;

solve;
display tot;
display x;

data;
set P := 1 2 3 4 5;
param : score crone :=
  1 1 6
  2 1.8 12
  3 1.4 10
  4 0.6 4
  5 1.4 8;
param B := 20;
end;
```

Subtask 3.c

The rounding heuristic applied to the solution x^* gives a feasible solution x' . Which one? With the knowledge collected until this stage which of the three following statements is correct:

- (i) x' is certainly optimal
- (ii) x' is certainly not optimal
- (iii) x' might be optimal

(Remember to justify your answer.)

Solution The rounding heuristic updates x^* setting $x_2 = 0$ or $x_2 = 1$. The latter gives an infeasible solution while the former gives $[1, 0, 0, 0, 1]$ with value 2.4. We cannot say at this stage if x' is optimal because the optimality gap 3.3–2.4 is not closed. Hence (iii) is correct.

Subtask 3.d

The two subproblems generated by the branch-and-bound algorithm after finding x^* correspond to choosing or not choosing a particular project. Which one?

Solution The solution is $[1, 0.5, 0, 0, 1]$ and the only fractional variable is x_2 hence we branch on it.

Subtask 3.e

Suppose the branch-and-bound algorithm considers first the subproblem corresponding to not choosing this project. Let's call this subproblem and its corresponding node in the search tree SP1. What is the optimal solution to its LP relaxation?

Solution Adding the constraint $x_2 \leq 0$ to the GLPK code above we obtain:

$$x = [1, 0, 0.2, 1, 1]$$

and $z = 3.3$.

Subtask 3.f

Next, the branch-and-bound algorithm considers the subproblem corresponding to choosing the project, i.e., subproblem SP2. Find the optimal solution to its LP relaxation. Which are the active nodes (i.e., open subproblems) at this point?

Solution Adding the constraint $x_2 \geq 1$ to the GLPK code above we obtain:

$$x = [0, 1, 0, 0, 1]$$

and $z = 3.2$. This is an integer solution and hence a lower bound.

Node SP2 is not active since an integer solution prunes the subtree. The other node SP1 has however still potential to find a better solution since its upper bound is $3.3 > 3.2$, hence the list of active nodes contains SP1.

Subtask 3.g

How does the branch and bound end?

Solution We need to examine the active nodes. Hence we branch once more with $x_3 \leq 0$ (subproblem SP3) and $x_3 \geq 1$ (subproblem SP4). The LP relaxation of SP3 gives an integer solution $[1, 0, 0, 1, 1]$ of value 3 and SP4 gives $[0.33, 0, 1, 0, 1]$ of value 3.13. Hence the upper bound from subtree SP1 is 3.13 which is smaller than the lower bound 3.2 of SP2 and we can prune SP4 by bounding. The optimal solution is the one on node SP2.

Task 4 (10%)

Some video camera must be allocated to watch the corridors of a museum. The graphs in Figure 6 represent the two floors of the museum, in which arcs correspond to corridors and vertices to the crossing points between corridors. A camera, allocated on a vertex, is able to watch all and only the arcs incident to the vertex. The problem in each floor consists in determining the minimal number of cameras and their position in order to watch all the corridors.



Figure 6: Two graphs representing the two floors of the museum of Task 4. On the left the first floor, on the right the second floor. [You find online the code of these graphs exhibiting the list of edges.]

Subtask 4.a

Formulate the problem

- for any general graph and
- for the two graphs representing the floors

in terms of integer linear programming.

Solution

The problem is a vertex covering problem. The formulation that we saw in class is the following:

$$\begin{aligned} \min \sum_{u \in V} x_u \\ x_u + x_v \geq 1 \quad \forall uv \in E \\ x_u \in \{0, 1\} \end{aligned}$$

A weak dual of the vertex covering problem is the maximum matching problem.

Subtask 4.b

What can you say about the possibility to solve the problems on the two floors using only linear programming? (Remember to justify your answers.)

Solution

In one of the exercises during the course whose answer can also be found in [B1] we saw that in the general case the vertex covering has fractional solutions. However the constraint matrix has only two entries different from zero in each column. We saw that such a matrix for undirected graphs is TUM if it is the incident matrix of a bipartite graph. A bipartite graph can be identified by the fact that it has no odd cycles. The graph on the left has no odd cycle, and it is indeed a bipartite graph with vertex partition $(1, 3, 4, 6, 9, 11)$ and $(8, 5, 7, 10, 2)$. The solution to the vertex cover will be integer. The graph on the right instead has an odd cycle hence it is not bipartite and we have no element to rule out the possibility of fractional solutions. Solving the problem however will give an integer solution.

```
x[1].val = 0
x[2].val = 1
x[3].val = 0
x[4].val = 0
x[5].val = 1
x[6].val = 0
x[7].val = 1
x[8].val = 1
x[9].val = 0
x[10].val = 1
x[11].val = 0
```

```
x[1].val = 0
x[2].val = 1
x[3].val = 0
x[4].val = 0
x[5].val = 1
x[6].val = 0
x[7].val = 1
x[8].val = 1
x[9].val = 1
x[10].val = 0
```

Task 5 (10%)

While you are holding this exam, the staff at IMADA will take a final decision about the creation of a new study area for students in the Mathematical Library. If the decision is positive some journals currently stored in the library will have to be moved in secondary facilities such as the basement or at the central Library. Each secondary storage facility has limited capacity and a particular access cost for retrieving information. Through appropriate data collection, made available by the diligent work of Tove, we can determine the usage rates for the information needs of the users. Let b_j denote the capacity storage of facility j and v_j denote the access cost per unit item from this facility. In addition, let a_i denote the number of items of a particular class i requiring storage and let u_i denote the expected rate (per unit of time) that we will need to retrieve books from this class. Our goal is to help Tove in deciding where to store the journals in a way that will minimize the expected retrieval cost.

Subtask 5.a

Show how to formulate the problem of determining an optimal policy as a network flow problem;

Subtask 5.b

Describe briefly an algorithm and explain the computational cost of its operations (only stating the total computational complexity without explanation is not enough).

Solution The problem is a transportation problem with data represented in Figure 7.

The problem can be solved by a min cost flow algorithm such as the cycle canceling algorithm. One first has to find a feasible flow, by solving a max flow problem. The flow has to saturate all source outgoing arcs. Then the flow is augmented through cycles

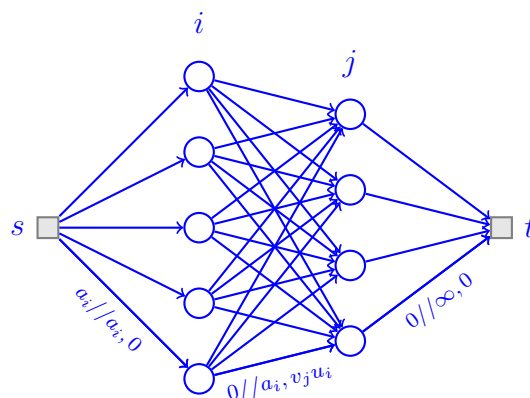


Figure 7:

of negative cost in the reduced network. It turns out, however, (Evans, 1984) that the following greedy algorithm is optimal: repeatedly assign items with the greatest retrieval rate to the storage facility with lowest access cost.

Task 6 (20%)

Consider the following LP problem [you find online its data in text format]:

$$\begin{array}{rll} \text{maximize} & 2x_1 + 3x_2 + 5x_3 + 3x_4 & \\ \text{st} & x_1 + 2x_2 + 3x_3 + x_4 \leq 5 & \\ & x_1 + x_2 + 2x_3 + 3x_4 \leq 3 & \\ & x_1, x_2, x_3, x_4 \geq 0 & \end{array}$$

and call x_5, x_6 the two slack variables to bring the problem in standard form.

Subtask 6.a

Using only matrix computations, like in the revised simplex method, calculate the value of the variables corresponding to a solution x^* with $x_B^* = (x_2, x_3)$ basic variables and $x_N^* = (x_1, x_4, x_5, x_6)$ non basic variables. [Hint to be faster in inverting the matrix A_B you can use any program for linear algebra. For example, in R, the inverse of a matrix can be calculated by `solve` and matrix multiplications by `%*%.`]

Solution From the theory we know that $x_B = A_B^{-1}b$. The matrices A_B for $x_B = (x_2, x_3)$ and A_N for $x_B = (x_1, x_4, x_5, x_6)$ are:

$$A_B = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix} \quad A_N = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 3 & 0 & 1 \end{bmatrix}$$

and its inverse computed in R with:

```
A=matrix(c(2,1,3,2),ncol=2,byrow=FALSE)
A1=solve(A)
A1%*%A # let's check
```

is

$$A_B^{-1} = \begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix}$$

Thus:

```
b=c(5,3)
A1%*%b
      [,1]
[1,]  1
[2,]  1
```

which gives $x_2 = 1, x_3 = 1$.

Subtask 6.b

Show that the solution found at the previous subtask is optimal but that it is not unique. Consequently, express all optimal solutions.

Solution To check whether this solution is optimal we need to compute the reduced costs \bar{c}_N of the non basic variables. From theory we need two calculations:

1. calculate the dual multipliers: $y = c_B A_B^{-1}$
2. $\bar{c}_N = c_N - y^T A_N$

Hence in R:

```
cN=c(2,3,0,0)
cB=c(3,5)
AN=matrix(c(1,1,1,3,1,0,0,1),ncol=4,byrow=FALSE)
y=cB%%A1
cN-y%%AN
      [,1] [,2] [,3] [,4]
[1,]    0  -1  -1  -1
```

Since all reduced costs are non-positive the solution is optimal. However one reduced cost is null. This is an indication that there are infinite optimal solutions. To find them we need to make a linear combination of the optimal vertices. Let's bring x_1 in basis. To decide which variable between x_2 and x_3 is going to leave the basis we need to compute how much we can increase them:

$$x'_B = x_B - A_B^{-1} A_N x_N x'_B = x_B - A_B^{-1} a t \geq 0$$

where a is the column of the variable entering in the basis.

```
> a=c(1,1)
> A1%%a
      [,1]
[1,]   -1
[2,]    1
```

$$x'_B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} t \geq 0 \quad \Rightarrow \quad \begin{matrix} t \geq -1 \\ t \leq 1 \end{matrix} \leftarrow$$

Hence the variable to leave the basis is x_3 . The new A_B matrix is:

$$A_B = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

and its inverse computed in R with:

```

> AB=matrix(c(2,1,1,1),ncol=2,byrow=FALSE)
> AB
      [,1] [,2]
[1,]    2    1
[2,]    1    1
> AB1=solve(AB)
> AB1
      [,1] [,2]
[1,]    1   -1
[2,]   -1    2

```

is

$$A_B^{-1} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

Thus:

```

b=c(5,3)
AB1%*%b
      [,1]
[1,]    2
[2,]    1

```

which gives $x_2 = 2, x_1 = 1$.

All the solutions can be expressed by linear combination of the two solutions found: $\alpha \vec{x}^* + (1 - \alpha) \vec{x}'$, $\alpha \geq 0$, that is,

$$\begin{aligned}
x_1 &= (1 - \alpha) \\
x_2 &= \alpha + (1 - \alpha)2 = 2 - \alpha \\
x_3 &= \alpha \\
x_4 &= 0 \\
x_5 &= 0 \\
x_6 &= 0
\end{aligned}$$

Subtask 6.c

Which minimal change in the data of the problem, maintaining all data integer, would make x^* become the unique optimal solution.

Solution Changing the cost coefficient of x_1 in the objective function from 2 to 1 would ensure that all reduced cost become negative and would not affect the feasibility of x^* , hence it would become the only optimal solution.

Subtask 6.d

Suppose we wish to consider a change in the right-hand side term of the second constraint, adding a value $\epsilon \in \mathbb{R}$.

- (i) For which values of ϵ would the variables x_2 and x_3 still form an optimal basis?
- (ii) In the interval determined at (i) would there still be multiple optimal solutions?
- (iii) In the interval determined at (i) how would the objective value change as a function of ϵ ?

Solution From the matrix representation of the simplex method we have $\bar{b} = A^{-1}b$ and $z = y^T b$. Hence,

$$\bar{b} = A^{-1}b = \begin{bmatrix} 2 & -3 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 3 + \epsilon \end{bmatrix} = \begin{bmatrix} 10 - 9 - 3\epsilon \\ -5 + 6 + 2\epsilon \end{bmatrix} \Rightarrow \begin{matrix} 1 - 3\epsilon \geq 0 \\ 1 + 2\epsilon \geq 0 \end{matrix} \Rightarrow \begin{matrix} \epsilon \leq 1/3 \\ \epsilon \geq -1/2 \end{matrix}$$

Hence for $\epsilon \in [-1/2, 1/3]$ the solution x^* remains feasible. Since in this case changing the b terms does not affect the reduced costs, then all previous optimal solutions remain optimal in that interval. The objective value can be calculated using the multipliers (here y). Hence:

$$\begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 3 + \epsilon \end{bmatrix} = 5 + 3 + \epsilon = 8 + \epsilon$$