Department of Mathematics and Computer Science
University of Southern Denmark, Odense

December 15, 2024
Marco Chiarandini

DM545/DM871 – Linear and integer programming

Sheet 7, Autumn 2024

Solution:

Included.

Exercises with the symbol $^+$are to be done at home before the class. Exercises with the symbol *will be tackled in class and should be at least read at home. The remaining exercises are left for self training after the exercise class.

Exercise 1$^+$

Consider the following three matrices:

$$\begin{bmatrix} 1 & 1 & -1 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix}$$

For each of them say if it is totally unimodular and justify your answer.

Solution:

We look for the satisfaction of the conditions of the theorem saw in class. Accordingly, it is *sufficient* for a matrix to be TUM to find a partition of the rows such that the ones with same sign are in different partitions and those with different sign in the same partition.
The first matrix is TUM. The partition is $I_1 = \{1, 4\}$ and $I_2 = \{2, 3\}$.
The second matrix is TUM. The partition is $I_1 = \{1, 2, 3\}$ and $I_2 = \{4\}$.
The third matrix is not TUM. Here the theorem does not apply and there is a submatrix with deteminant -2, hence we cannot be sure that the solutions associated with the matrix will be integer.

Exercise 2$^+$

1. In class we stated that for the uncapaciteted facility location problem there are two formulations:

$$X = \{(x, y) \in \mathbb{R}^m_+ \times \mathbb{B}^1 : \sum_{i=1}^{m} x_i \leq my, x_i \leq 1 \text{ for } i = 1, \ldots, m\}$$

$$P = \{(x, y) \in \mathbb{R}^m_+ \times \mathbb{R}^1 : x_i \leq y \text{ for } i = 1, \ldots, m, y \leq 1\}$$

Prove that the polyhedron $P = \{(x_1, \ldots, x_m, y) \in \mathbb{R}^{m+1} : y \leq 1, x_i \leq y \text{ for } i = 1, \ldots, m\}$ has integer vertices. [Hint: start by writing the constraint matrix and show that it is TUM.]

Solution:

For $m = 1..5$ the constraint matrix of P is:

$$\begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & y \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \end{array} \begin{matrix} \leq 0 \\ \leq 0 \\ \leq 0 \\ \leq 0 \\ \leq 0 \\ \leq 0 \end{matrix}$$

Its transpose is:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

This matrix satisfies the requirements for the sufficient condition of totally unimodularity seen in class. Hence, P has integer vertices and is the convex hull description of the points that are feasible solutions to the uncapacitated facility location problem.

2. Consider the following (integer) linear programming problem:

$$\begin{aligned} \min \quad & c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 \\ & x_3 + x_4 \geq 10 \\ & x_2 + x_3 + x_4 \geq 20 \\ & x_1 + x_2 + x_3 + x_4 \geq 30 \\ & x_2 + x_3 \geq 15 \\ & x_1, x_2, x_3, x_4 \in \mathbb{Z}_0^+ \end{aligned} \tag{1}$$

The constraint matrix has consecutive 1's in each column. Matrices with consecutive 1's property for each column are totally unimodular. Show that this fact holds for the specific numerical example (1). That is, show first that the constraint matrix of the problem has consecutive 1s in the columns and then that you can transform this matrix into one that you should recognize to be a TUM matrix. [Hint: rewrite the problem in standard form (that is, in equation form) and add a redundant row $\mathbf{0} \cdot \mathbf{x} = 0$ to the set of constraints. Then perform elementary row operations to bring the matrix to a TUM form.]

Solution:

Let's write the concatenated matrix of the system of linear equations that we get from the equational standard form:

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 10 \\ 0 & 1 & 1 & 1 & 0 & -1 & 0 & 0 & 20 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & 0 & 30 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 15 \end{bmatrix}$$

Adding a row filled with zeros does not affect the solutions of the linear system:

$$A' = \begin{bmatrix} 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 10 \\ 0 & 1 & 1 & 1 & 0 & -1 & 0 & 0 & 20 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & 0 & 30 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We can then perform elementary row operations that also preserve the solutions. In particular, on each row $i = 5, 4, 3, 2$ we subtract the row $i - 1$ obtaining:

$$A'' = \begin{bmatrix} 0 & 0 & 1 & 1 & -1 & 0 & 0 & 0 & 10 \\ 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 10 \\ 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 10 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & -1 & -15 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & -15 \end{bmatrix}$$

The matrix now satisfies the requirements for the sufficient condition of totally unimodularity that we saw in class. Moreover, in this particular case the b terms sum to zero and the problem has acquired the same structure as a minimum flow problem and can therefore be solved by the network simplex.

Exercise 3$^+$

In class, we proved that the (mininum) vertex covering problem and the (maximum) matching problem are a weak dual pair. Prove that for bipartite graphs they, actually, are a strong dual pair.

Solution:

The formulation of the matching problem is:

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e & & \\ & \sum_{e \in E : v \in e} x_e \leq 1 & & \forall v \in V \\ & x_e \in \{0,1\} & & \forall e \in E \end{aligned}$$

If we take the linear relaxation and make the dual of it we obtain:

$$\begin{aligned} \min \quad & \sum_{v \in V} y_v & & \\ & y_v + y_u \geq w_{uv} & & \forall u, v \in V, uv \in E \\ & y_v \geq 0 & & \forall v \in V \end{aligned}$$

This latter is the linear relaxation of the vertex cover problem.
Hence the two problems make a weak dual pair. It is not strong, indeed a triangle has optimal vertex cover 2 and optimal matching 1.
In bipartite graphs instead the pair is strong dual. Indeed, the solution of the linear relaxation of the matching problem on bipartite graphs is always integer. The same must old for the dual and hence the gap is closed.
A more formal proof is on page 31-32 and ch. 4 of [Wo].

Exercise 4* MILP Modeling

Shift scheduling. The administrators of a department of a urban hospital have to organize the working shifts of nurses maintaining sufficient staffing to provide satisfactory levels of health care. Staffing requirements at the hospital during the whole day vary from hour to hour and are reported in Table 1.

Hour	Staffing requirement
0 am to 6 am	2
6 am to 8 am	8
8 am to 11 am	5
11 am to 2 pm	7
2 pm to 4 pm	3
4 pm to 6 pm	4
6 pm to 8 pm	6
8 pm to 10 pm	3
10 pm to 12 pm	1

Table 1:

According to union agreements, nurses can work following one of the seven shift patterns in Table 2 each with its own cost.

pattern	Hours of work	total hours	cost
1	0 am to 6 am	6	720 Dkk
2	0 am to 8 am	8	800 Dkk
3	6 am to 2 pm	8	740 Dkk
4	8 am to 4 pm	8	680 Dkk
5	2 pm to 10 pm	8	720 Dkk
6	4 pm to 12 pm	8	780 Dkk
7	6 pm to 12 pm	6	640 Dkk

Table 2:

The department administrators would like to identify the assignment of nurses to working shifts that meets the staffing requirements and minimizes the total cost.

1. Model the problem as a MILP problem.

Solution:

$$
\begin{array}{rrrrrrrrrrrrrrrrr}
\min & 720x_1 & + & 800x_2 & + & 740x_3 & + & 680x_4 & + & 720x_5 & + & 780x_6 & + & 640x_7 & \\
0-6: & x_1 & + & x_2 & & & & & & & & & & & \geq 2 \\
6-8: & & & x_2 & + & x_3 & & & & & & & & & \geq 8 \\
8-11: & & & & & x_3 & + & x_4 & & & & & & & \geq 5 \\
11-14: & & & & & & & x_4 & + & x_5 & & & & & \geq 7 \\
14-16: & & & & & & & & & x_5 & + & x_6 & & & \geq 3 \\
16-18: & & & & & & & & & x_5 & + & x_6 & + & x_7 & \geq 4 \\
18-20: & & & & & & & & & x_5 & + & x_6 & + & x_7 & \geq 6 \\
20-22: & & & & & & & & & x_5 & + & x_6 & + & x_7 & \geq 3 \\
22-24: & & & & & & & & & & & x_6 & + & x_7 & \geq 1 \\
\multicolumn{17}{c}{x_1, x_2, x_3, x_4, x_5 \leq 0 \text{ and integer}}
\end{array}
$$

2. Use the results from Exesrcise 2 to show that the *shift scheduling problem* can be solved efficiently when formulated as a mathematical programming problem. (You do not need to find numerical results.)

Solution:

The constraint matrix has the consecutive ones property on columns, hence it is TUM and the problem can be solved by linear programming.

Exercise 5*

Generalized Assignment Problem. Suppose there are n types of trucks available to deliver products to m clients. The cost of truck of type i serving client j is c_{ij}. The capacity of truck type i is Q_i and the demand of each client is d_j. There are a_i trucks for each type. Formulate an IP model to decide how many trucks of each type are needed to satisfy all clients so that the total cost of doing the deliveries is minimized. If all the input data will be integer, will the solution to the linear programming relaxation always be integer?

Solution:

It is good trying to model the problem as a min cost flow problem. However, this is not possible. The number of trucks can be seen as the flow but there are capacity restrictions on the flows complicating the situation. So we cannot model the problem as a min cost flow. We can however write an ILP model:

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \tag{2}$$

$$\sum_{j \in J} x_{ij} \leq a_i \qquad \forall i \in I \tag{3}$$

$$\sum_{i \in I} Q_i x_{ij} \geq d_j \qquad \forall j \in J \tag{4}$$

$$x_{ij} \geq 0 \text{ and integer} \qquad \forall (i,j) \in A \tag{5}$$

The solutions of the linear relaxation are not necessarily integer, becasue this is not a min cost flow model and the matrix is not trivially TUM.

Exercise 6* Network Flows: Problem of Representatives

A town has r residents $R_1, R_2, \ldots, R_r$; q clubs $C_1, C_2, \ldots, C_q$; and p political parties $P_1, P_2, \ldots, P_p$. Each resident is a member of at least one club and belongs to exactly one political party. Each club must nominate one of its members to represent it on the town's governing council so that the number of council members belonging to the political party P_k is at most u_k. Is it possible to find a council that satisfies this "balancing" property?

Show how to formulate this problem as a maximum flow problem.

Solution:
These exercise is taken from [AMO] pages 170-176.

Assumption 6.5. *The network does not contain parallel arcs (i.e., two or more arcs with the same tail and head nodes).*

This assumption is essentially a notational convenience. In Exercise 6.24 we ask the reader to show that this assumption imposes no loss of generality.

Before considering the theory underlying the maximum flow problem and algorithms for solving it, and to provide some background and motivation for studying the problem, we first describe some applications.

6.2 APPLICATIONS

The maximum flow problem, and the minimum cut problem, arise in a wide variety of situations and in several forms. For example, sometimes the maximum flow problem occurs as a subproblem in the solution of more difficult network problems, such as the minimum cost flow problem or the generalized flow problem. As we will see in Section 6.6, the maximum flow problem also arises in a number of combinatorial applications that on the surface might not appear to be maximum flow problems at all. The problem also arises directly in problems as far reaching as machine scheduling, the assignment of computer modules to computer processors, the rounding of census data to retain the confidentiality of individual households, and tanker scheduling. In this section we describe a few such applications; in Chapter 19 we discuss several other applications.

Application 6.1 Feasible Flow Problem

The feasible flow problem requires that we identify a flow x in a network $G = (N, A)$ satisfying the following constraints:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b(i) \qquad \text{for } i \in N, \tag{6.2a}$$

$$0 \le x_{ij} \le u_{ij} \qquad \text{for all } (i, j) \in A. \tag{6.2b}$$

As before, we assume that $\sum_{i\in N} b(i) = 0$. The following distribution scenario illustrates how the feasible flow problem arises in practice. Suppose that merchandise is available at some seaports and is desired by other ports. We know the stock of merchandise available at the ports, the amount required at the other ports, and the maximum quantity of merchandise that can be shipped on a particular sea route. We wish to know whether we can satisfy all of the demands by using the available supplies.

We can solve the feasible flow problem by solving a maximum flow problem defined on an augmented network as follows. We introduce two new nodes, a source node s and a sink node t. For each node i with $b(i) > 0$, we add an arc (s, i) with capacity $b(i)$, and for each node i with $b(i) < 0$, we add an arc (i, t) with capacity $-b(i)$. We refer to the new network as the *transformed network*. Then we solve a maximum flow problem from node s to node t in the transformed network. If the maximum flow saturates all the source and sink arcs, problem (6.2) has a feasible solution; otherwise, it is infeasible. (In Section 6.7 we give necessary and sufficient conditions for a feasible flow problem to have a feasible solution.)

It is easy to verify why this algorithm works. If x is a flow satisfying (6.2a)

and (6.2b), the same flow with $x_{si} = b(i)$ for each source arc (s, i) and $x_{it} = -b(i)$ for each sink arc (i, t) is a maximum flow in the transformed network (since it saturates all the source and the sink arcs). Similarly, if x is a maximum flow in the transformed network that saturates all the source and the sink arcs, this flow in the original network satisfies (6.2a) and (6.2b). Therefore, the original network contains a feasible flow if and only if the transformed network contains a flow that saturates all the source and sink arcs. This observation shows how the maximum flow problem arises whenever we need to find a feasible solution in a network.

Application 6.2 Problem of Representatives

A town has r residents $R_1, R_2, \ldots, R_r$; q clubs $C_1, C_2, \ldots, C_q$; and p political parties $P_1, P_2, \ldots, P_p$. Each resident is a member of at least one club and can belong to exactly one political party. Each club must nominate one of its members to represent it on the town's governing council so that the number of council members belonging to the political party P_k is at most u_k. Is it possible to find a council that satisfies this "balancing" property?

We illustrate this formulation with an example. We consider a problem with $r = 7$, $q = 4$, $p = 3$, and formulate it as a maximum flow problem in Figure 6.1. The nodes $R_1, R_2, \ldots, R_7$ represent the residents, the nodes $C_1, C_2, \ldots, C_4$ represent the clubs, and the nodes $P_1, P_2, \ldots, P_3$ represent the political parties.

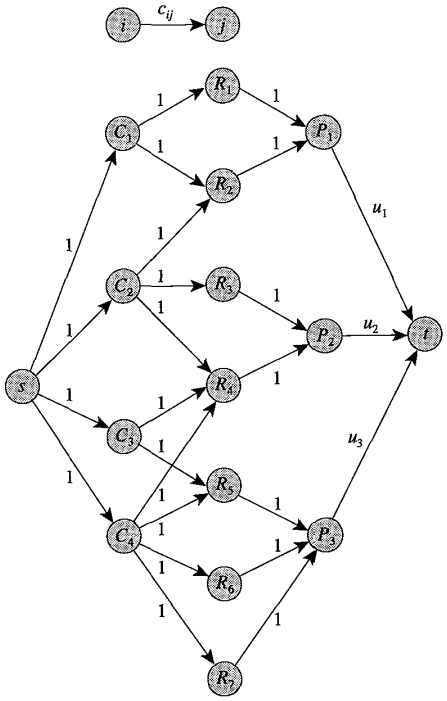

Figure 6.1 System of distinct representatives.

The network also contains a source node s and a sink node t. It contains an arc (s, C_i) for each node C_i denoting a club, an arc (C_i, R_j) whenever the resident R_j is a member of the club C_i, and an arc (R_j, P_k) if the resident R_j belongs to the political party P_k. Finally, we add an arc (P_k, t) for each $k = 1, \ldots, 3$ of capacity u_k; all other arcs have unit capacity.

We next find a maximum flow in this network. If the maximum flow value equals q, the town has a balanced council; otherwise, it does not. The proof of this assertion is easy to establish by showing that (1) any flow of value q in the network corresponds to a balanced council, and that (2) any balanced council implies a flow of value q in the network.

This type of model has applications in several resource assignment settings. For example, suppose that the residents are skilled craftsmen, the club C_i is the set of craftsmen with a particular skill, and the political party P_k corresponds to a particular seniority class. In this instance, a balanced town council corresponds to an assignment of craftsmen to a union governing board so that every skill class has representation on the board and no seniority class has a dominant representation.

Application 6.3 Matrix Rounding Problem

This application is concerned with consistent rounding of the elements, row sums, and column sums of a matrix. We are given a $p \times q$ matrix of *real* numbers $D = \{d_{ij}\}$, with row sums α_i and column sums β_j. We can round any real number a to the next smaller integer $\lfloor a \rfloor$ or to the next larger integer $\lceil a \rceil$, and the decision to round up or down is entirely up to us. The matrix rounding problem requires that we round the matrix elements, and the row and column sums of the matrix so that the sum of the rounded elements in each row equals the rounded row sum and the sum of the rounded elements in each column equals the rounded column sum. We refer to such a rounding as a *consistent rounding*.

We shall show how we can discover such a rounding scheme, if it exists, by solving a feasible flow problem for a network with nonnegative lower bounds on arc flows. (As shown in Section 6.7, we can solve this problem by solving two maximum flow problems with zero lower bounds on arc flows.) We illustrate our method using the matrix rounding problem shown in Figure 6.2. Figure 6.3 shows the maximum flow network for this problem. This network contains a node i corresponding to each row i and a node j' corresponding to each column j. Observe that this network

				Row sum
	3.1	6.8	7.3	17.2
	9.6	2.4	0.7	12.7
	3.6	1.2	6.5	11.3
Column sum	16.3	10.4	14.5	

Figure 6.2 Matrix rounding problem.