Operations Research
TD n°2

Master Computer Science 1st year
Université Nice-Sophia Antipolis

1 Maximum Flow modeling

These exercises come from "Network Flows: Theory, Algorithms, and Applications", Ahuja, Ravindra K. and Magnanti, Thomas L. and Orlin, James B. 1993, Prentice-Hall, Inc.

1.1 System 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 $P_1, P_2, \ldots, P_s$. 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?

1.2 Matrix Bounding 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.

				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	

1.3 Scheduling on Uniform Parallel Machines

In this application we consider the problem of scheduling of a set J of jobs on M uniform parallel machines. Each job $j \in J$ has a processing requirement p_j (denoting the number of machine days required to complete the job), a release date r_j (representing the beginning of the day when job j becomes available for processing), and a due date $d_j \geq r_j + p_j$ (representing the beginning of the day by which the job must be completed). We assume that a machine can work on only one job at a time and that each job can be processed by at most one machine at a time. However, we allow preemptions (i.e., we can interrupt a job and process it on different machines on different days). The scheduling problem is to determine a feasible schedule that completes all jobs before their due dates or to show that no such schedule exists. Scheduling problems like this arise in batch processing systems involving batches with a large number of units. The feasible scheduling problem, described in the preceding paragraph, is a fundamental problem in this situation and can be used as a subroutine for more general scheduling problems, such as the maximum lateness problem, the (weighted) minimum completion time problem, and the (weighted) maximum utilization problem.

Job (j)	1	2	3	4
Processing time (p_j)	1.5	1.25	2.1	3.6
Release time (r_j)	3	1	3	5
Due date (d_j)	5	4	7	9

1.4 Tanker Scheduling Problem

A steamship company has contracted to deliver perishable goods between several different origin-destination pairs. Since the cargo is perishable, the customers have specified precise dates (i.e., delivery dates) when the shipments must reach their destinations. (The cargoes may not arrive early or late.) The steamship company wants to determine the minimum number of ships needed to meet the delivery dates of the shiploads. To illustrate a modeling approach for this problem, we consider an example with four shipments; each shipment is a full shipload with the characteristics shown in Figure. For example, as specified by the first row in this figure, the company must deliver one shipload available at port A and destined for port C on day 3. Figure also shows the transit times for the shipments (including allowances for loading and unloading the ships) and the return times (without a cargo) between the ports.

Ship-ment	Origin	Desti-nation	Delivery date
1	Port A	Port C	3
2	Port A	Port C	8
3	Port B	Port D	3
4	Port B	Port C	6

(a)

	C	D
A	3	2
B	2	3

(b)

	A	B
C	2	1
D	1	2

(c)