OPTIMIZATION ALGORITHMS M

Part 1 - Exercises

Exercise 1

Given n "items" and a "container", a "weight" p_j and a "cost" c_j (with p_j and c_j positive integers) are associated with each item j (j = 1, …, n).

Determine a subset M of the n items so that:

a) the sum of the weights of the items in M is not smaller than a given value a;

b) the cardinality of M is not smaller than a given value b;

c) the sum of the costs of the items in M is minimum.

1) Prove that the problem is NP-hard.
2) Define a Linear Integer Programming model for the considered problem.
3) Define the complexity of the problem for determining a feasible solution for the following problems:

3.1) original problem;

3.2) problem with constraint a) imposed, and constraint b) replaced by the constraint imposing that the cardinality of M must be equal to b;

3.3) problem with constraint a) imposed, and constraint b) replaced by the constraint imposing that the cardinality of M is not greater than b;

3.4) problem with constraint b) imposed, and constraint a) replaced by the constraint imposing that the sum of the weights of the items of M must be equal to a.

Exercise 2

Given n "operations" and m "machines", an "initial time" a_j and a "final time" b_j are associated with each operation j ($j = 1, \ldots, n$) . Each machine can process at any time at most one operation, and can globally work for a time period not greater than a given value C.

1) Prove that the problem for determining a feasible solution of the considered problem is NP-hard.
2) Define a Linear Integer Programming model for the considered problem in the case in which the number of used machines must be minimized.

Exercise 3

Given a "depot" which must serve m "customers". The customers can be served by using n different "routes". In particular, each customer i (i = 1, …, m) can be served by a subset V_i of routes (with V_i contained in the set {1, 2, …, n}). Each route j (j = 1, …, n) has a "cost" c_j and a "traveling time" t_j (with c_j e t_j non-negative). Determine a subset S of the n routes such that:

a) each customer is served by at least one route of S;
b) the sum of the traveling times of the routes of S is not smaller than a given value d;
c) the sum of the costs of the routes of S is minimum.

1) Define a Linear Integer Programming model for the considered problem.
2) Prove that the problem is NP-hard.
3) Define the complexity of the problem for determining a feasible solution for the considered problem.
4) As at point 3) in the case in which in constraint b) it is imposed that the sum of the traveling times of the routes of S is equal to a given value d

Exercise 4

Given m "items" and n "vehicles": a positive "weight" p_j is associated with each item j (j= 1, , m); a positive "capacity" a_i is associated with each vehicle i (i = 1, …, n). Also assume: m > n > 0.

Determine the items to be loaded into the vehicles so that:

a) the sum of the weights of the items loaded into each vehicle i is not greater than the capacity a_i ;
b) each item j is loaded into no more than one vehicle;
c) the global number of items loaded into the vehicles is smaller than a given value k;
d) the sum of the weights of the items loaded into the vehicles is maximum.

1) Define a Linear Integer Programming model for the considered problem.
2) Prove that the problem is NP-hard.
3) Define the complexity of the problem for determining a feasible solution for the following problems:
   3.1) original problem;
   3.2) problem with "not smaller" instead of "smaller" in constraint c);
   3.3) problem with "equal" instead of "smaller" in constraint c).

Exercise 5

Given a "directed graph" $G = (V, A)$, with $| V | = n$ and $| A | = m$. A positive "cost" $c_{i,j}$ is associated with each arc (i, j) in A. Assume also that the vertex set V is partitioned into K subsets ("regions") $R_1, R_2, \ldots, R_K$, with $R_1 = \{1\}$.

Determine an "elementary circuit" of G (i.e., a circuit passing at most once through each vertex of G) visiting at least one vertex of each of the K regions, and such that the sum of the costs of the arcs of the circuit is minimum.

1)- Prove that the problem is NP-hard.

2)- Define a Linear Integer Programming model for the considered problem.

3)- Assuming that the graph G is complete and that the cost matrix $(c_{i,j})$ satisfies the "triangularity condition" (i.e.: $c_{i,k} + c_{k,j} <= c_{i,j}$ for each triple (i, j, k) of vertices of V), define the new Linear Integer Programming model so as to "strengthen" the constraints of the model.

Exercise 6

Given a "complete directed graph" G = (V, A), with | V | = n. A positive "cost" $c_{i,j}$ (with $c_{i,i}$ = infinity for each vertex i of V) is associated with each arc (i, j) of A. A positive "prize" p_i is associated with each vertex i of V

Given a vertex r of V, determine an "elementary circuit" of G visiting vertex r and such that:

a) the sum of the costs of the arcs of the circuit is minimum;
b) the sum of the prizes associated with the vertices of the circuit is not smaller than a given value a;
c) the number of vertices of the circuit having a prize smaller than a given value b is not smaller than a given percentage d of the number of vertices of the circuit (with d between 0 and 1).

1)- Define a Linear Integer Programming model for the considered problem.

2)- Prove that the problem is NP-hard. (determine two NP-hard problems reducible to the considered problem).

3) Define the complexity of the problem for determining a feasible solution for the considered problem.

4)- Define the complexity of the problem in the case in which: $c_{i,j}$ = K for each arc (i, j) of G (with K given value).

Exercise 7

Given n "jobs" and m "machines". The "cost" per processing job j (j = 1, …, n) on machine i (i = 1, …, m) is given by $c_{i,j}$, while the corresponding "processing time" is given by $t_{i,j}$. If machine i (i = 1, …, m) is used, there is an additional cost equal to b_i (this cost is equal to zero if machine i is not used). The values $c_{i,j}$, $t_{i,j}$ and b_i are positive integers.

Assign each job to one and only one machine so that:

a) the global processing time for each machine i (i = 1, …, m) is not greater than a given value a_i (with a_i positive and integer);

b) the "global" cost is minimum.

1)- Define the complexity of the problem for determining a feasible solution for the considered problem.

2)- Prove that the considered problem is NP-hard.

3)- Define a Linear Integer Programming model for the considered problem, so as to minimize the number of constraints.

4)- Define additional Linear Integer Programming models which, by using a larger number of constraints, can produce "lower bounds" (obtained with the continuous relaxation of the model) better than those which can be obtained with the continuous relaxation of the model defined at point 3).

Exercise 8

Given a "complete directed graph" $G = (V, A)$, with $| V | = n$: a "weight" $p_{i,j}$ and a non-negative "time" $t_{i,j}$ are associated with each arc (i, j) of A. Two disjoint subsets S and T are also given (with S and T contained in A).

Determine a "Hamiltonian circuit" H of G so that:

a) the sum of the weights of the arcs of H is maximum;

b) the sum of the times of the arcs of H is not greater than a given value d;

c) the number of arcs of H belonging to subset S is not smaller than the number of arcs of H belonging to subset T.

1)- Define a Linear Integer Programming model for the considered problem.

2)- Prove that the considered problem is NP-hard.

3) Define the complexity of the problem for determining a feasible solution for the considered problem.

4)- Define the complexity of the problem for determining a feasible solution for the problem in the case in which constraint b) is not imposed.

Exercise 9

Given n “depots” and m “customers”: each customer i (i = 1, …, m) has a non-negative “potential profit” p_i. Each depot j (j = 1, …, n) has a non-negative “cost” c_j and is able to “serve” a subset of the m customers. In particular, a binary matrix ($a_{i,j}$) is given, such that for each pair [depot j , customer i] (with j = 1, …, n and i = 1, …, m) $a_{i,j}$ = 1 if depot j is able to serve customer i, and $a_{i,j}$ = 0 otherwise.

For each subset S of the n depots, the corresponding “global profit” is given by the difference: (sum of the profits of the customers which can be served by the depots of S) - (sum of the costs of the depots of S).

Determine a subset S* of the n depots so that:

a) S* contains at most d depots (with d given value greater than 0 and smaller or equal to n);
b) the global profit of S* is maximum;
c) the total cost of the depots of S* is not smaller than a given non-negative value b.

1)- Define a Linear Integer Programming model for the considered problem.

2)- Prove that the considered problem is NP-hard.

3) Define the complexity of the problem for determining a feasible solution for the considered problem.

4)- Define the complexity of the problem for determining a feasible solution for the problem in the following cases:

4.1) subset S* must contain exactly d depots;

4.2) the global cost of the depots of subset S* is equal to b.

Exercise 10

Given a "complete directed graph" G = (V, A), with | V | = n. A non-negative "time" $t_{i,j}$ (with $t_{i,i}$ = infinity for each vertex i of V) is associated with each arc (i, j) in A (with the times satisfying the triangularity condition). A positive profit a_i is associated with each vertex i of V.

Given a vertex h of V, determine an "elementary circuit" of G visiting h and such that:

a) the number of arcs of the circuit is not greater than a given value r (with r between 2 and n);
b) the sum of the times of the arcs of the circuit is not greater than a given value d;
c) the sum of the profits associated with the vertices of the circuit is maximum.

1)- Define a Linear Integer Programming model for the considered problem.

2)- Prove that the considered problem is NP-hard.

3) Define the complexity of the problem for determining a feasible solution for the considered problem.

4) Define a Linear Integer Programming model for the variant of the considered problem in which the objective function (to be maximized) is given by:

alpha * (sum of the profits associated with the vertices of the circuit) –
beta * (maximum time of the arcs of the circuit)

with alpha and beta non-negative given values.