Part II of the Assignment

This is the second part of the final assignment for the course "Large Scale Optimization using Decomposition" (42136), Spring 2022.

Very important:

THIS ASSIGNMENT SHOULD BE WRITTEN INDIVIDUALLY.

Each participant **must** individually hand in a report and the required julia programs.

Furthermore:

- This assignment is handed out to all course participants 28/4, after the lecture.
- Hand-in deadline is 12/5, 12.00, on DTU-Learn of both report (as pdf) and julia programs.
- Most of the data (but not all) is available in the data file: "optigas_data.txt", which you can also download from DTU-Learn.
- The assignment consists of 11 questions, (Q1 - Q11). The answer to the assignment is a report, uploaded as a pdf file, answering these 11 questions, **and** working julia programs for the questions: Q1, Q4, Q6, Q7 and Q10. The julia programs **must** also be uploaded to DTU-Learn.

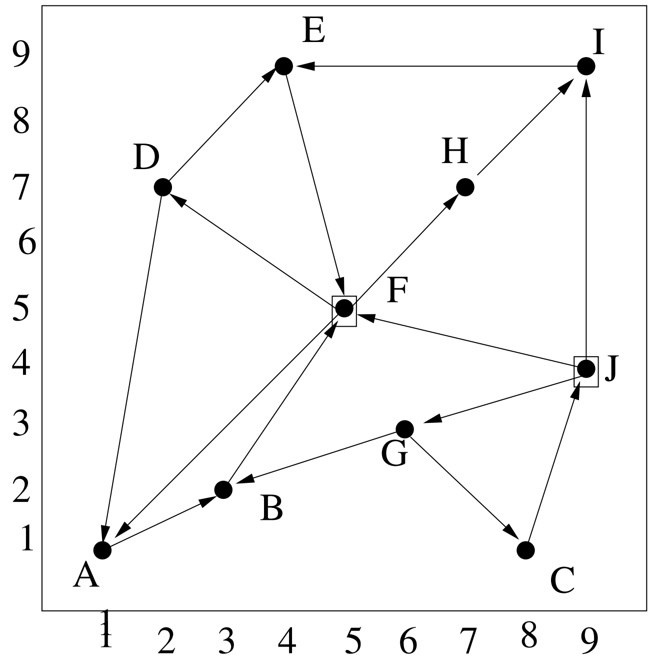

Figure 1: The available gas network

OptiGas

After successfully graduating as an Engineer from DTU, you have immediately been hired as a consultant at the firm Benders Consultants (BC). BC has been contacted by the company OptiGas. OptiGas is a natural-gas distribution company, who is considering expanding their network into a new region of Denmark. OptiGas wants you to give quantitative decision support about the cost of the expansion.

Question 1: OptiGas consider to buy another company, who already **has** a gas pipeline network in the region. This network is given in Figure 1. It consists of 10 nodes (each of which represents a location in a smaller city), A to J. In two of these cities, F and J, gas can be fed into the network (from OptiGas's central network), these are the gas-production nodes. The remaining cities each have a demand for gas. The position of the nodes is given as (x,y) positions. The arrows betwween the nodes, correspond to the gas pipelines. In a pipeline, the gas can only be sent one way, as indicated by the direction of an arc. The cost of sending gas between the cities corresponds directly to the distance (the cost for the pumping gas). New pipelines can be built, but this is obviously very expensive, here set to 10 times the distance between the two nodes. The data are described in detail on the last page, and the data file can be downloaded from DTU Inside.

Formulate a MIP model to optimize the gas flow and the possible expansion of the network with new pipelines (arcs) assuming that OptiGas has bought the other company and hence has the network in Figure 1 available (this is also in the fixed cost data). Assume Optigas want to minimize the total cost.

Upload julia program with name **Q1.jl**

You observe that Benders Decomposition can be applied to the MIP in Q1.

Question 2: Formulate the Benders subproblem.

Question 3: Formulate the Benders master problem.

Question 4: Implement the Benders algorithm. As the initial solution for the integer variables, select arcs in the existing network at value one and do not select any other arcs. Report the number of iterations, upper bound and lower bound (in the end).

Upload julia program with name **Q4.jl** and a log file **Q4.log** with a line for each iteration with master objective, sub objective and the upper bound.

Question 5: OptiGas are now considering to build their own gas-network from scratch and will not buy the other company. Hence it needs to be able to start without an initial network. This will lead to unbounded subproblems in the Benders algorithm. Formulate the Ray-generation problem.

Question 6: Implement the Benders algorithm, now including the ray-generation problem. Initially, assume that no new gas-pipelines are to be built.

	A	B	C	D	E	F	G	H	I	J
1	5	35	25	0	50	0	20	30	25	0
2	5	35	25	10	30	0	20	30	35	0
3	25	45	25	0	50	0	10	20	15	0
4	5	25	35	0	50	0	20	30	25	0
5	5	35	35	0	50	0	10	20	35	0
6	5	35	25	0	60	0	20	20	25	0
7	5	25	15	0	60	0	20	40	25	0
8	5	25	25	0	60	0	30	30	15	0
9	5	35	25	10	30	0	20	30	35	0
10	5	35	15	0	50	0	20	50	15	0

Table 1: Scenario Information

Upload julia program with name **Q6.jl** and a log file **Q6.log** with a line for each iteration with master objective, sub/ray objective and the upper bound.

Question 7: OptiGas now learns that you have worked with stochastic programming during your education. They already know that the **demand** is stochastic, so OptiGas' market-analysis group create a scenario analysis, with 10 scenarios of the expected gas demand. These scenarios are given in Table . Each row corresponds to a scenario.

Assuming that there is an equal likelihood for the different scenarios, i.e. each will occur with a probability of $\frac{1}{10}$, formulate the deterministic-equivalent, now with demand dependent on the scenarios. (Notice that the demand in the first scenario is the same as that which was used in the previous questions). Notice this table is **not** part of the datafile, you have to write it into the Julia program yourself.

Upload julia program with name **Q7.jl**

Question 8: Formulate the Benders subproblem for this stochastic program.

Question 9: Formulate the Benders master problem for this stochastic program.

Question 10: Implement the Benders algorithm with ray generation for this stochastic program. Initially, assume that no new gas-pipelines are to be built.

Upload julia program with name **Q10.jl** and a log file **Q10.log** with a line for each iteration with master objective, sub objective and the upper bound.

Question 11: How can OptiGas' problem (without scenarios) be solved using Dantzig-Wolfe decomposition and Column Generation/Branch & Price. Here, no implementation is necessary, but describe your suggestion including LP and MIP models of the master and subproblems, using at most 3 pages.

FAQ

1. **Which subproblem can/should we use in Question 3 and the non-ray Benders algorithm?** : In Question 3 you can use either the dual subproblem (u variables) or the non-dualized sub-problem (x, z) variables). Notice however, that the ray-generation problem (problem 5) can only be solved in the dual version.

2. **Can we use the CPLEX automatic benders decomposition algorithm?** : We will allow the use of CPLEX automated Benders algorithm in Question 6 and Question 10. No other place, and it does not lead to full points.

3. **Is it an advantage to implement the Benders algorithm in the callback version?** : Yes, but only in Question 6 and Question 10. If you implement and upload a working Benders algorithm using a callback, we will give extra points.

The data-file (you can download from DTU Inside: ”optigas_data.txt”) :

```
#----------------------------------------------------------------------
# PARAMETERS
Nodes     = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "j"]
N=length(Nodes)
   node_x = [ 1    3    8    2    4    5    6    7    9    9 ]
   node_y = [ 1    2    1    7    9    5    3    7    9    4 ]
gas_arc = [
              0    1    0    0    0    0    0    0    0    0;
              0    0    0    0    0    1    0    0    0    0;
              0    0    0    0    0    0    0    0    0    1;
              1    0    0    0    1    0    0    0    0    0;
              0    0    0    0    0    1    0    0    0    0;
              1    0    0    1    0    0    0    1    0    0;
              0    1    1    0    0    0    0    0    0    0;
              0    0    0    0    0    0    0    0    1    0;
              0    0    0    0    1    0    0    0    0    0;
              0    0    0    0    0    1    1    0    1    0
]

distance=zeros(Float64,N,N)
for n1=1:N
    for n2=1:N
        distance[n1,n2]=floor(sqrt( (node_x[n1]-node_x[n2])*(node_x[n1]-node_x[n2]
                          (node_y[n1]-node_y[n2])*(node_y[n1]-node_y[n2]) ) )
    end
end
fixed_cost=zeros(Float64,N,N)
for n1=1:N
    for n2=1:N
        fixed_cost[n1,n2]=10*distance[n1,n2]#*(1-gas_arc[n1,n2])
    end
end
F=10*sum( gas_arc[i,j]*distance[i,j] for i=1:N,j=1:N)
gas_production= [  0 0 0 0 0 95 0 0 0 95 ]
gas_demand= [  5 35 25 0 50 0 20 30 25 0 ]
#----------------------------------------------------------------------
```

Brief comments:

- Nodes: The names of the nodes (cities)
- N: Number of cities
- node_x and node_y: position of node
- gas_arc: Incidence matrix: gas_arc[i,j]=1 if there is a gas pipline from i to j
- gas_production: The amount of gas injected into the network at city i (in thousands of cubic-meters)
- gas_demand: The amount of gas subtracted from the network at city i (in thousands of cubic-meters)
- distance: Distance between cities, euclidian distance, based on coordinates. This is the transport price pr. thousand of cubic-meters of gas)
- fixed_cost: Cost for installing a pipeline between cities