Homework 1, LINMA 2450
Nicolas Stevens
25/10/2022

1 The shortest path problem

In this homework, you will study different ways to solve the *shortest path problem*. Let's consider a *directed* graph $D = (V, A)$ with arc lengths $c_{ij} \geq 0$ (assuming the two nodes $i, j \in V$ are connected by an arc $e = (ij) \in A$). Taking two distinct nodes $s, t \in V$, the shortest path problem consists of finding a shortest path from s to t.

Your are asked to solve *four instances*, all of them with *three different algorithms* implemented in Julia (see section 2). The four instances are attached to this assignment (in .json files[1]) and are named *small*, *medium*, *large1* and *large2*. Those are dictionaries containing the following fields:

- `N`: the list of the nodes.
- `From`, a dictionary that reads as follows: `From[i]` contains the list of nodes accessible from `i`, i.e. the list of nodes j such that the graph includes an edge (i, j) (in other words, the edges of the graph are $e = (i, j)$, $\forall i \in N, j \in From(i)$).
- `To`, a dictionary that reads as follows: `To[i]` contains the list of nodes that can reach `i`, i.e. the list of nodes j such that the graph includes an edge (j, i).
- `weight[i][j]`: the length of the edge (i, j).
- `s`: the label of the starting node for the shortest path.
- `t`: the label of the target (end) node for the shortest path.

The three algorithms you are asked to implement are:

Question 1 (*Solving the IP with a Commercial solver such as Gurobi*)

(a) Write the IP model of the shortest path problem. Define clearly your variables and briefly explain the model.

(b) Implement the IP model in Julia (JuMP) and solve the four instances with Gurobi. Do not forget to turn off automatic *heuristic*, *presolve*, and *cut generation* — detailed explanation in the next page.

[1] Those can be read using e.g. `JSON.parsefile("small.json")`

Question 2 (*Convex Hull formulation*)

(a) Write the convex hull of the shortest path problem. Prove it is the convex hull. Explain how to leverage this result to solve the shortest path problem.

(b) Implement the convex hull formulation in Julia and use it to solve the four instances.

Question 3 (*Dynamic programming*)

(a) Describe a DP algorithm that solves the shortest path problem. Give the recurrence formula and characterise its complexity (be accurate on how you make your computations *and how you store the intermediate results*).

(b) Implement the DP algorithm and solve the four instances.

Report content and code format You will have to submit both your code and a written report (max 3 pages). The code of the three algorithms should be written in Julia and should, for each algorithm, output (i) the solution (optimum value of the variables), (ii) the associated objective and (iii) the run time. In your report, you are asked to:

- Provide the requested answers to the above questions.
- Report the results (i), (ii) for the four instances (for the variables, don't report all of them, only the selected edges). Also report result (iii) (the run time) for the four instances, for the three different algorithms.
- Briefly comment those results.

2 Julia

In this semester, the students need to use Julia for implementation. Julia is an open source high-level programming language optimized for numerical analysis and computations. A domain specific language JuMP embedded in Julia is designed specifically for mathematical programming. JuMP is fast, solver independent, and has a syntax structure which mimics natural mathematical expressions. Starting from the USA, the Julia community is growing fast especially for mathematical programming community.

You need to prepare four things before start coding.

- Julia Language
- JuMP Manual
- Optimization Solver (Gurobi, CPLEX, etc) Gurobi / Gurobi JuMP Manua
- Choose your favourite IDE (e.g. Atom, Jupyter notebook, etc.).

For the optimization solver, you can use something else, but we highly recommend Gurobi. You can get a student license for free.

3 Parameter Settings for Gurobi

For comparing different IP formulations numerically, it is important to turn off some settings of the solver. For Gurobi, you need to set the following three parameters to zero: Presolve, Heuristics, and Cuts. One way to set the parameters in Gurobi is using set optimizer attribute (`set_optimizer_attribute( myModel, "Presolve", 0)`, `set_optimizer_attribute( myModel, "Heuristics", 0)`, `set_optimizer_attribute( myModel, "Cuts", 0)`). You can check the details in the Gurobi JuMP Manual above.

4 Submission instructions

This homework should be performed by groups of two students. You should submit both code and report in a zip file until 23:59 24/11/2022. No late acceptance. The report and the code should be the *personal* work of each group. This assignment will account for 1 point out of the total 20 points for this course.

- Code (in julia)
- Report (.pdf file within 3 pages)
- File Name (include FirstName LastName for each group members)