COSC 364
Internet Technologies and Engineering
Flow Assignment Description

Andreas Willig
Dept. of Computer Science and Software Engineering
University of Canterbury
email: andreas.willig@canterbury.ac.nz

January 19, 2024

---

1 Administrivia

This assignment is part of the COSC 364 assessment process. It involves formulating an optimization problem, writing a program (in Python3) to generate an LP file for this problem and using CPLEX to get numerical solutions.

It is quite likely that the problem description below leaves a lot of things unclear to you. Please do not hesitate to use the "Question and Answer Forum" on the LEARN platform for raising and discussing any unclear issues. **Important:** Please do not email me with technical questions, rather send such questions to the LEARN forum, so that all students can benefit.

Note that this assignment description refers to version 1.0 of the planning booklet.

2 Plagiarism Warning

Your submissions are logged and originality detection software will be used to compare your solution with other solutions (from this year and from previous years). Dishonest practice, which includes

- letting someone else create all or part of an item of work,
- copying all or part of an item of work from another person with or without modification, and

- allowing someone else to copy all or part of an item of work,

may lead to partial or total loss of marks, no grade being awarded and other serious consequences including notification of the University Proctor.

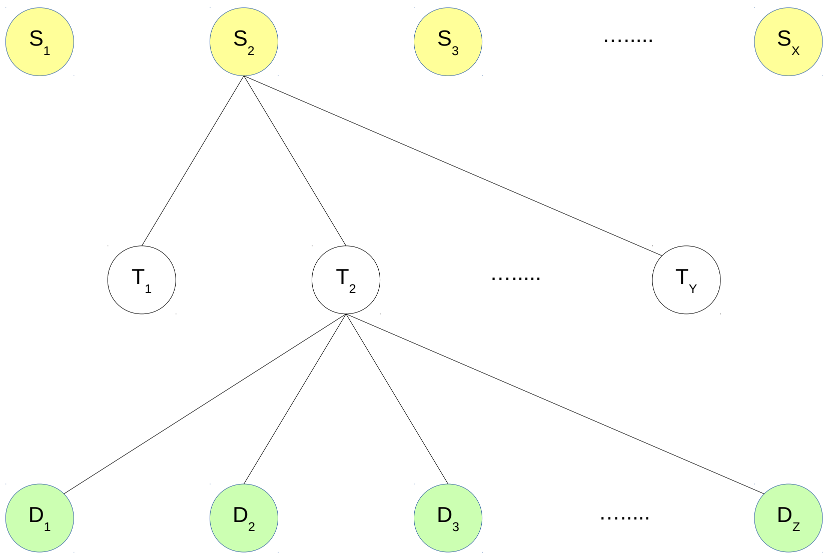
You are encouraged to discuss the general aspects of a problem with others. However, anything you submit for credit must be entirely your own work and not copied, with or without modification, from any other person. **Your source code and text must be completely your own**, any sharing of source code with other groups / persons is expressly forbidden. If you need help with specific details relating to your work, or are not sure what you are allowed to do, contact your tutor or lecturer for advice. If you copy someone else's work or share details of your work with anybody else, you are likely to be in breach of university regulations and the Computer Science and Software Engineering department's policy. For further information please see

- Academic Integrity Guidance for Staff and Students
  `www.canterbury.ac.nz/ucpolicy/GetPolicy.aspx?file=Academic-Integrity-Guidance-For-Staff-And-Students.pdf`
- Academic Integrity and Breach of Instruction Regulations in the University Calendar
  `www.canterbury.ac.nz/regulations/general-regulations/academic-integrity-and-breach-of-instruction-regulations/`

3 Problem Description

Broadly, the problem builds on the lab problems described in Section 7.4 of the planning booklet in Version 1.0, in particular problems 7.4.1 and 7.4.2. Furthermore, it is also related to problem 5.2.6.

We are given a network with the following structure:



There are X source nodes denoted as $\{S_1, S_2, \ldots, S_X\}$, Y transit nodes denoted as $\{T_1, T_2, \ldots, T_Y\}$ and Z destination nodes denoted as $\{D_1, D_2, \ldots, D_Z\}$. A source node has links to all transit nodes, and a transit node has links to all destination nodes. A source then has as many paths available towards a destination as there are transit nodes. Instead of using the δ_{kpl} notation introduced in Section 5.1.2 of the planning booklet it might be easier to use decision variables of the form x_{ikj}, referring to the part of the demand volume between source node i and destination node j that is routed through transit node k.

The capacities of the links have to be determined. We denote these capacities as follows:

- For a link between source node S_i and transit node T_k we denote its capacity by c_{ik}.
- For a link between transit node T_k and destination node D_j we denote its capacity by d_{kj}.

Furthermore, between source node S_i $(1 \leq i \leq X)$ and destination node j $(1 \leq j \leq Z)$ there is a demand volume of

$$h_{ij} = i + j$$

units. There is a global requirement that **each demand volume shall be split over exactly two different paths**, such that each path gets an equal share of the demand volume. In other words, if we denote by x_{ikj} the amount of flow for the demand volume between source i and destination j that uses the path through transit node k, then x_{ikj} must be positive for exactly two values k_1, k_2 (i.e. $x_{ik_1j} > 0$ and $x_{ik_2j} > 0$) and must be zero for all other $k \in \{1, \ldots, Y\}$.

With this background, solve the following tasks:

- The objective is to balance the load (i.e. the total amount of incoming traffic flow) on all the transit nodes. Formulate an optimization problem for generic values of X, Y and Z (with $Y \geq 2$), subject to the usual constraints and the additional requirement that each demand volume should be split over exactly two paths. Determine the load on the transit nodes, the capacities of all links and the value of each flow. Please give a mathematical formulation (showing the objective function, the decision variables and all constraints) and explain it carefully.
- Write a program in Python3 which accepts three positive integer numbers X, Y and Z as input and which generates a valid LP file for the above problem.
- Fix $X = 7$ sources and $Z = 7$ destinations. Run your program for each $Y \in \{3, 4, 5, 6, 7\}$, solve the resulting LP file with CPLEX and record the following outputs:
  - The CPLEX run time on your computer (under Linux the `time` command is handy).
  - The load on the transit nodes.
  - The capacity of the link with the highest capacity.

4 Deliverable

Each student submits a **single** `.pdf` file, which includes answers to the questions given below. The file needs to include:

- A title page listing name and student-id.
- A section showing your problem formulation and your explanation of it.
- A section showing the results for the CPLEX execution time, the number of links with non-zero capacities, the lowest and highest load of the transit nodes, and the highest-capacity links for varying Y. Please explain your results. This explanation should **not** be a verbal description of **what** we see, but discuss **why** we see what we see.
- The source code of your program as an appendix.
- A generated LP file (for $X = 3$, $Y = 2$ and $Z = 3$) as an appendix.

Warning:

- Reports that are not in pdf format or which contain more than one file automatically receive 0 marks!!
- Please make sure that your `.pdf` file contains all fonts and can be completely printed on UC printers – if it cannot be printed it will **not be marked**.
- By submitting, you declare that you have read and understood the plagiarism warning in this assignment description and that your submitted work is entirely yours. Should we find evidence that you have engaged in plagiarism-related activities, you will automatically receive 0 marks for the assignment, and your name will be submitted to the University Discipline Register!!

The report has to be submitted via the LEARN page for COSC364. You find the submission deadline and an assignment submission box in the Assessment section on LEARN. Late submissions are **not** accepted, unless approved through the UC special consideration process.

5 Marking

The marking will be based on the deliverable only, and the main components are the correctness of the shown results, the problem formulation, the explanation of the problem formulation and the results, and the correctness of the included LP file.

When after a quick glance I find that your source code is particularly ugly and unorganized, I will apply deductions of up to 10% of the achievable marks, but otherwise you will not receive any extra marks for the source code.