

# COSC 364

## Internet Technologies and Engineering

### Second Assignment

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

April 5, 2018

---

## 1 Administrivia

This assignment is part of the COSC 364 assessment process. It is worth 15% of the final grade. It involves formulating an optimization problem, writing a program (in Python, Java or C) 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 Pair Work

An important sub-goal of this assignment to give you some pair-work experience. The rules around this are as follows:

- You are expected to work in groups of two persons, submissions of larger groups will not be marked.
- It is your responsibility to find a partner, but I offer assistance (see below).
- An individual submission comes with an automatic penalty of 15%, unless there is exactly one individual submission. No exception other than

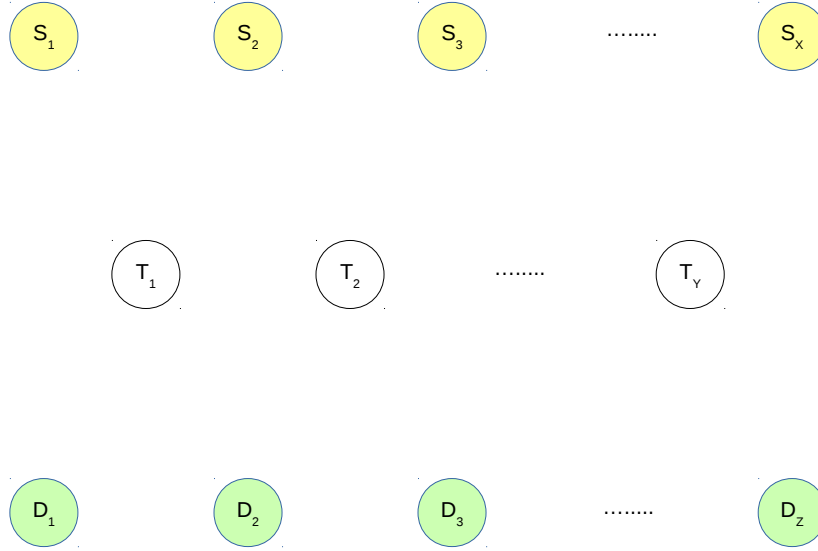
those related to “special consideration” will be given to this rule. Clearly, sometimes things may not work out well with a partner. An important piece of advice in this context is to **start early** with the assignment, so that you have enough time to find a new partner, should that happen.

If you do not find a partner on your own (you could try to post on the **learn** forum!), then you can send me an email, stating the programming languages that you can possibly work in. I will then try to find a suitable partner for you from those who sent similar emails.

### 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  $\{S_1, S_2, \dots, S_X\}$ ,  $Y$  transit nodes denoted as  $\{T_1, T_2, \dots, T_Y\}$  and  $Z$  destination nodes denoted as  $\{D_1, D_2, \dots, D_Z\}$ . 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  $\delta_{kpl}$  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 three 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 three values  $k_1, k_2, k_3$  (i.e.  $x_{ik_1j} > 0$ ,  $x_{ik_2j} > 0$  and  $x_{ik_3j} > 0$ ) and must be zero for all other  $k \in \{1, \dots, Y\}$ .

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  $Y \geq 3$ ), subject to the usual constraints and the additional requirement that each demand volume should be split over exactly three 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 either Java, Python or C, 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 pair of students submits a **single .pdf file** which includes answers to the questions given below. It is sufficient when one member of a team submits this. The file needs to include:

- A title page listing name and student-id of both partners, and indicating the percentage contribution of each partner. **Note:** this must be agreed upon by you and your partner, the relative weights will influence grading.
- A section showing your problem formulation and your explanation for this.
- A section showing the results for the CPLEX execution time, the number of links with non-zero capacities and the highest-capacity links for varying  $Y$ . Please explain your results.
- The source code of your program as an appendix.
- A generated LP file (for  $X = 3$ ,  $Y = 2$  and  $Z = 4$ ) as an appendix. The restriction to  $Y = 2$  here is intentional to keep the size of the listing limited.

**Note:** 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**. Files not submitted in `.pdf` format will **not be marked**!

The documentation has to be submitted via the `learn` page for COSC364. The submission deadline is **Friday, May 25, 2018, 11.59pm**. 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 and 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 un-organized, 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.

## References