Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Resilient Networks

Exercise Sheet: Network Design

Prof. Dr. Mathias Fischer, August See

Goals and Objectives Recapitulate the lecture on network design. After covering so many optimization problems in the lecture, it is time for you to gather some practical experience with an optimization tool and to solve some network design problems.

Tool The task requires you to use the IBM ILOG CPLEX Optimization Studio[1], which is available for free at the IBM website for students and academics. In case you want to obtain the tool, you can register with an university email address and you can download the tool afterwards. Here, we are using version 12.8.

IBM ILOG CPLEX Optimization Studio is an Eclipse-IDE for the Optimization Programming Language (OPL), which is, as the name implies, a programming language for optimization problems. Once installed, you can start the IDE on Linux by executing the following path to the standard installation folder of the program:

```
/opt/ibm/ILOG/CPLEX_Studio128/opl/oplide/oplide
```

You find detailed instructions on OPL and help on the Internet on quite a number of different websites. A good introduction is the following video blog and the OPL user manual:

- Video Tutorials: [2]
- IBM Knowledge Center [3]
- OPL Language Reference Manual [4]

[1] https://www.ibm.com/products/ilog-cplex-optimization-studio
[2] https://mediacenter.ibm.com/tag/tagid/ibmilogcplexoptimizationstudio
[3] https://www.ibm.com/support/knowledgecenter/SSSA5P_12.8.0/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html
[4] https://www.ibm.com/support/knowledgecenter/SSSA5P_12.8.0/ilog.odms.studio.help/pdf/opl_langref.pdf

Preparing Linux VM The following only applies if you want to use the provided VM in moodle. Download the installer `cplex_studio128.linux-x86-64.bin` from moodle and save it inside your VM. Open a terminal in the same folder and run the installer:

```
$ chmod +x cplex_studio128.linux-x86-64.bin
$ sudo ./cplex_studio128.linux-x86-64.bin
```

Follow the instructions on the screen and wait for the installer to finish. Run the IBM ILOG CPLEX Optimization Studio using the command above.

Submission Solve this task sheet in groups as you organized yourselves beforehand. Your solution should be handed in as a pdf document that describes your results and how you got there. In addition, submit your code that solves the tasks. In particular, your solution should document the following:

- A description of your approach and methodology towards the solution
- A summary and interpretation of your results (figures, tables, examples...)
- The code that solves the tasks

Submit your group solution to this task sheet in Moodle. Ask any open question during the Q&A Session or in the forum.

1 Loading an Example Project

Load the provided example project *min-link-cost* into the OPL-IDE by extracting the zip archive and importing the project as existing OPL project into the IDE. Make yourself familiar with the IDE and study the provided example.

You will notice that the example consists out of

- one file for the code (*min-link-cost.mod*),
- one file for the data (*min-link-cost.dat*) as input to the optimization problem,
- and one file containing the settings for the project (*min-link-cost.ops*).

The example denotes a network taken from the lecture according to Figure 1 and represents an optimization problem that tries to find a minimum link capacity assignment given a fixed set of demands and paths. The notation of formulas in OPL is similar to the one we used in our lecture.

Start the solver for the optimization problem by pressing on the green start button as shown in Figure 2. The solution tab on the bottom will then display the results.

1. Explain in your words which information the variable `cpaths` holds. How is it related to the demands and paths?

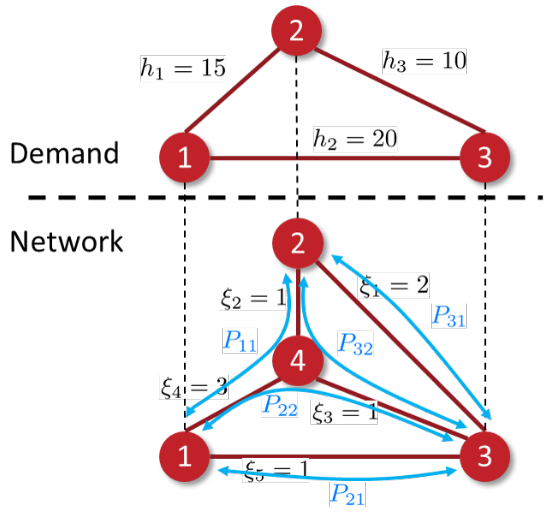


Figure 1: Example taken from the lecture.

2. What is the variable `cpaths` used for in the constraints? Why is the information hold by this variable required in the formulas?

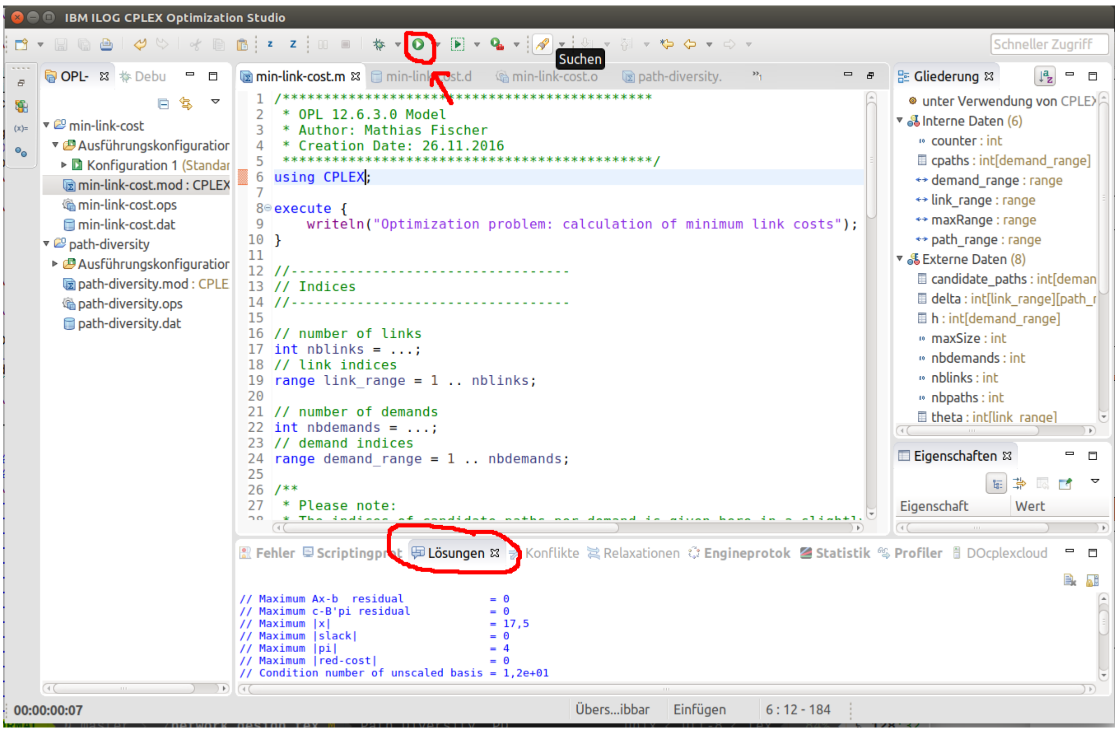
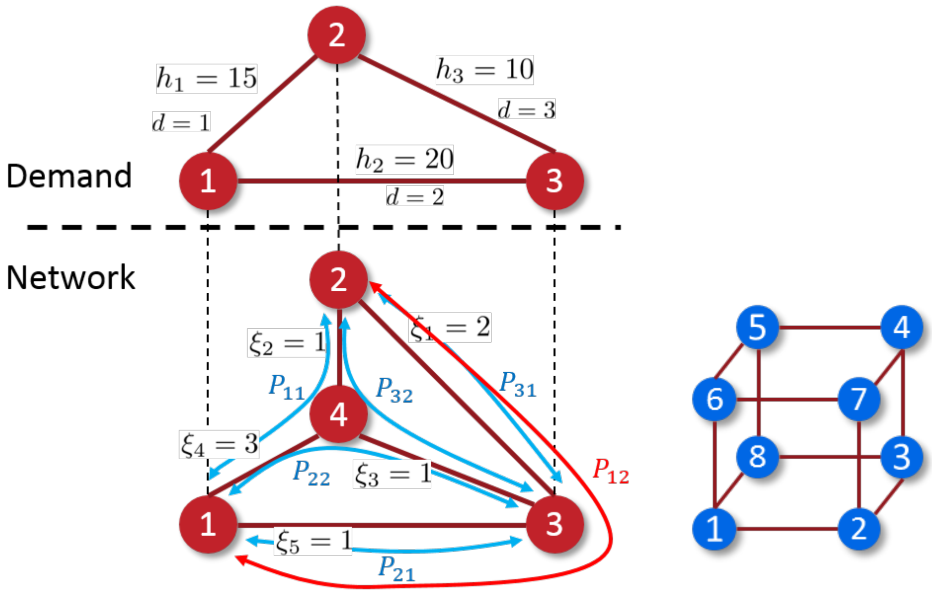


Figure 2: IBM ILOG CPLEX IDE.

3. What is the outcome of the calculation? Use the values of the decision variables in the solution to answer the initial network design problem.

2 Adding a Path

Modify the presented example to include an additional path for the first demand. For that, add a path P_{12} via node 3 as given in Figure 3(a). Recompute the solution to the optimization problem.



(a) Modified example from lecture with additional path P_{12} marked in red

(b) Cube network

Figure 3: 3(a) network modified by adding one more path and 3(b) a cube network.

1. Apart from modifying the variables `candidate_path` and `nbpaths`, you need to update δ_{edp} (`delta`). How is the data structure `delta` interpreted? Consequently, how is `delta` extended to include the additional path?

2. Can you see a difference in the solution compared to the results in Task 1? Explain why the format of the decision variables changed. Also, explain why the new solution to the network design problem seems reasonable compared to the previous result.

3 Path Diversity (PD)

The network in Figure 3(a) has two paths per demand. Hence, it would be a good idea to enhance the example, so that every demand is satisfied by splitting the demand flow evenly in between two paths (diversity factor $n_d = 2$). Enhance the given optimization problem accordingly and compute the result.

1. Based on previous results without path diversity, explain your expectations regarding a solution to this problem. Think of potential values for the result. With the current model and especially the decision variables, are your expected values a possible solution? We assume that demands can be divided into smaller flows of arbitrary size, i.e., flows are not required to be in whole numbers. What needs to be changed?

2. Implement required changes to the model and calculate the decision variables. Give an interpretation why this solution provides path diversity.

4 Unrestricted Demand Reconfiguration (DR-U)

In the lecture we also covered demand flow restoration schemes. Take the network from Figure 3(a), so that an unrestricted demand reconfiguration (DR-U) in case of errors is done. Compute the dimensioning of the link capacities and a demand flow assignment for the given paths.

1. What is missing in the current model to consider DR-U? How to incorporate DR-U into the model? Explain the changes for the implementation of your solution.

2. Interpret your results and why your solution to the network design problem satisfies DR-U.

5 Optional: Expert Tasks

To strengthen your practical experience even more, we suggest you to solve the following optional tasks.

5.1 Cube Network

Revisiting the path diversity problem from Task 3, now replace the network from Figure 3(a) by the cube network from the lecture as given in Figure 3(b). All neighboring nodes have a pairwise demand of $\forall d \in 1, \ldots, D : h_d = 3$ and the path costs are identical to the hop count ($\forall e \in 1, \ldots, E : \xi_e = 1$). Compute a path diversity solution with a diversity factor $n_d = 3$.

1. How is your δ_{edp} (`delta`) constructed for the cube network. Is there anything more in the model or data you have to modify?
2. Interpret your results with respect to the network design problem.

5.2 A Design Problem of Your Choice

Take another protection or restoration (LP) design problem from the lecture, implement it, and compute a solution.