Energy-efficiency vs. resilience
Code for optimization procedures and simulations

P. Chołda and P. Jaglarz

To reproduce the steps of the algorithm presented in the paper, save attached files to a common directory and provide required tools.

CONTENTS

The authors are with the AGH University of Science and Technology, Department of Telecommunications, Krakow, Poland. E-mail: {piotr.cholda}@agh.edu.pl.

SOFTWARE REQUIREMENTS

- CPLEX (including OPL Interpreter): http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
- MATLAB (including MATLAB Compiler): http://www.mathworks.com/products/matlab/

RISK-OPTIMIZATION.MOD

This CPLEX script presents the optimization problem described in the paper. It finds the optimal mix of recovery options assigned to various connections when a given risk mitigation strategy is assumed.

```
// ********************************************/
// OPL 12.6.0.0 Model
// Author: Piotr Cholda, AGH University of Science and Technology
// Creation Date: 21 May 2014 at 15:03:30
// ********************************************/

float coefficient_energy = ...; // monetary equivalent of a unit of capacity

float Energy_baseline = ...; // energy for provisioning non recovered connections

float coefficient_risk = ...; // monetary cost of a unit of risk

float Risk_baseline = ...; // the level of risk for non recovered connections

{string} RecoveryMethod = ...; // the used recovery method (in fact, t = {NR,DP,DL,SP,DL})

{string} Nodes = ...; // network nodes

tuple arc // network links (edges)
{
  string source;
  string destination;
}

{arc} Arcs with source in Nodes, destination in Nodes = ...;

tuple demand // demands (connections)
{
  string source;
  string destination;
}

{demand} Demands with source in Nodes, destination in Nodes = ...;

float risk[RecoveryMethod][Demands] = ...; // predicted value of risk incurred for demand d ↵
    ↳when it uses recovery method t with a given risk measure and compensation policy

float energy_usage[RecoveryMethod][Demands] = ...; // share in the energy usage for demand d ↵
    ↳when it uses recovery method t

dvar float+ Total_energy; // total energy used in the network

dvar float+ Involved_budget; // the monetary cost of providing additional energy for risk ↵
    ↳mitigation

dvar float+ Total_risk; // total risk in the network (expressed in monetary units)

dvar float+ Risk_decrease; // decrease of risk in comparison to the baseline risk

dvar boolean recovery_method[RecoveryMethod][Demands]; // = 1 if demand d uses recovery ↵
    ↳method t, = 0, otherwise

// Profit maximization:
minimize Involved_budget + Total_risk; // should be uncommented only if the profit ↵
    ↳maximization strategy is assumed

// Risk minimization and cost balance:
//minimize Total_risk; // should be uncommented only if the risk minimization or cost balance↵
    ↳ strategy is assumed

// Total benefit coverage:
//maximize Risk_decrease; // should be uncommented only if the total benefit coverage ↵
    ↳strategy is assumed

subject to{
```

```
59
60    forall(d in Demands)
61       sum(rm in RecoveryMethod) recovery_method[rm][d] == 1; // a demand uses only one recovery
          ↪ method
62
63    Total_energy == sum(rm in RecoveryMethod,d in Demands) coefficient_energy*energy_usage[rm][d
          ↪]*recovery_method[rm][d];
64
65    Involved_budget == Total_energy - coefficient_energy*Energy_baseline;
66
67    Total_risk == sum(rm in RecoveryMethod, d in Demands) coefficient_risk*risk[rm][d]*
          ↪recovery_method[rm][d];
68
69    Risk_decrease == coefficient_risk*Risk_baseline - Total_risk;
70
71    // Total benefit coverage:
72    //Involved_budget <= Risk_decrease; // should be uncommented only if the total benefit
          ↪coverage strategy is assumed
73
74    // Cost balance:
75    //Involved_budget <= Total_risk; // should be uncommented only if the cost balance strategy
          ↪is assumed
76
77  }
```