A metaheuristic for security budget allocation in utility networks

A metaheuristic for security budget allocation in utility networks

Jochen Janssens, Luca Talarico, Kenneth Sörensen

University of Antwerp Operations Research Group ANT/OR

Prinsstraat 13, 2000 Antwerp, Belgium

May 2015

Real-life utility networks such as smart grids, pipelines and water networks can be exposed to safety and security related risk. To mitigate the risks that might result in service interruptions for the users of these networks, countermeasures can be applied. In this paper, a decision model is proposed that assumes that all edges (e.g., pipes, cables) and nodes (e.g., switching or connection stations, substations in an electricity network) have a certain probability of failing, which can be reduced by applying appropriate security strategies. An optimization model is developed that determines the optimal security strategy to apply for each node and each arc so as to minimize the probability for disconnected node pairs to arise in the network, subject to a budget constraint. A metaheuristic approach to solve this problem is proposed. The metaheuristic is calibrated in a statistical experiment and detailed experiments on realistic instances confirm that it performs well.

Keywords: network security, metaheuristics, countermeasures, local search.

1 Introduction

People make the assumption that utility networks such as the ones used to transport electricity, water, gas, and data, are always available. They take for granted that a high level of service is guaranteed at all times and that the network can handle whichever load is placed on it. In reality, failures are not rare events. The increased dependency on continuous supply of commodities, such as water and power, makes today's industrialised society and economy much more vulnerable to supply interruptions. In addition, since modern countries are not used to having even short power blackouts, even a minor service failure can generate a large negative impact in the user's daily lives.

In some situations, power network component failures can have a significant impact on neighbouring segments, triggering cascading effects in the whole network. As a result of such an

event, service levels can therefore drop significantly, and supply can be temporarily interrupted. This might result in a disconnection of a user or even of certain subsections from the rest of the networks, affecting buildings, city block, a complete city, or even an entire regional district, or large areas of a country depending on the gravity of the outage. If a blackout spreads across borderlines, which is more likely today due to the interconnection of power grids between different countries, the impact can escalate as a function of the duration of the interruption (?).

In general, the majority of network failures has a duration of a few minutes up to a few hours. Nevertheless, in rare cases, major blackouts, affecting e.g., the electricity network or the water network, may last days or even weeks, significantly affecting private users, companies and critical infrastructure such as telecommunication networks, financial services and hospitals. For example, a technical failure in a substation in Uttar Pradesh (India) in 2001 cut 226 million people from the electricity grid for a long period of time, resulting in economic losses in the order of USD 110m, causing trains to be stranded for 15 hours, and crippling utilities and hospitals across the region (?).

Network interruptions can have different causes:(i) *safety*-related causes, such as construction defects, material failures, ground movement, natural disasters, unintentional human errors (ii) *security*-related causes, such as intentional terrorist acts, vandalism, sabotages, etc.

In this paper, we will focus on intentional actions aimed at damaging a utility network. It has been frequently demonstrated that utility networks represent vulnerable targets to terrorists. Major metropolitan areas and even multi-state regions could be subject to simultaneous attacks. Moreover, network segments located in isolated areas offer little resistance to attack and it might require several hours before they are fully restored.

In theory, terrorist attacks could strike a utility network at any time and in different modalities, since many elements of the network represent a potential target. Considering an electricity network, substations present the most vulnerable points. Transportation lines are even easier to be destroyed since they can be attacked anywhere along the line, but they are also easier to repair. Power generating station are the most difficult targets for attack since they are usually guarded and protected more effectively. A recent sabotage action in a Belgian nuclear power plant causing it to shut down for several months, proves that even well guarded places with an ample amount of security and safety regulations can be a viable target.

The goal of this paper is to develop a decision model to reduce the vulnerability of a utility network. In this model, each segment (arc or node) of the network has a certain probability to be targeted by intentional attackers and failing as a result of that attack. It is assumed that these probabilities are known or can be estimated. The manager of the utility network can implement countermeasures on each segment that reduce the likelihood of that segment failing. Each countermeasure has a certain cost, and the manager has a budget that cannot be exceeded. The aim of the model is to minimize the total risk of a network failure, which is defined as a situation in which there is no more functional connection between a pair of nodes in the network.

This paper is organized as follows. In Section 2 the literature on utility network security-related problems is briefly discussed. In Section 3, the problem of selecting the best strategies to increase the security of the whole network is described and modelled as an optimization problem. In Section 4, a metaheuristic to solve the network security problem is proposed. **??** presents the results of the heuristic on realistic instances. **??** discusses the main research findings and concludes the paper.

2 Literature review

The literature on securing utility networks from intentional attackers has attracted the attention of many researcher in the last years. Many papers deal with the problem of designing more robust and reliable utility networks in a preliminary stage, in which a network designer tries to decrease the probability of service failures by increasing the number of redundant links in the network. However, only few papers have addressed the problem of supporting decision makers to protect an existing utility network from external attacks by investing in effective security measures, subject to a budget constraint and technical limitations.

As mentioned, this paper proposes a decision model to support utility managers in the selection of cost-effective network protection investements. A similar approach has been proposed by **?** for the field of IT. In this paper, security countermeasures are used to guarantee the confidentiality, availability, and integrity of data in computer systems that might be subject to cyber-attacks. A decision support system, based on a genetic algorithm, is proposed to select the "best" combination of countermeasures that respects the user's preferred trade-off between the cost of the selected security measures and the resulting risk exposure of the computer system.

Considering a telecommunication network, a mixed integer programming approach is proposed by **?** to support decision makers in the selection of the optimal mix of countermeasures to prevent or mitigate cyber-threats in IT systems. The proposed model relies on the definition of a limited number of potential attack scenarios, which simplifies the decision process aimed at balancing expected worst-case losses and the cost of the selected security measures.

In **?** a model to assess the risk exposure of each segment of a pipeline network, is proposed. For each pipeline segment the risk exposure is computed considering the features of that segment and the hypothetical consequences of an accident scenario.

In **?** an approach aimed at formulating recommendations to more effectively protect a chemical cluster against existing systemic risks and decrease interdependent risks within chemical industrial areas, is proposed. A multi attribute method is developed modelling chemical industrial areas as interconnected and complex networks and using a holistic optimization approach considering inter-organisational and inter-cluster objectives. The goal of the model is to use a quantitative approach to map systemic risks in chemical industrial areas. Using the outcome of the model risk, experts and decision makers can make a quantitative risk assessment to objectivly inform them selves about possible prevention measures to lower the risks exposure.

However, after having analysed the network topology, the selection of the most appropriate safety measures is left to the decision maker.

The approach suggested here extends and generalize the existing works in the literature on securing utility networks by defining a single-objective problem and proposing a quantitative method to select appropriate security strategies. An objective function, which relies on the minimization of the probability of the network to be unavailable between any couple of network nodes, is used instead of the maximization of the effectiveness of the security measures used as done in **?**. Moreover, in this paper, since a list of security strategies is defined for each arc and each node of the network, the model incorporates not only decisions taken at the level of the network, as done in **?**, **?** and **?**, but it depends on the choices made at the level of single network arcs or single network nodes.

In a previous model of **?**, service interruptions, as a consequence of failures in a network, have been treated considering only two nodes, i.e., the point from which the service or the product is send to the customer and the customer or the point to which the product or service is delivered through the network. This simplified model enables network providers and managers to reduce the probability of a network breakdown by applying security strategies to arcs, that might be a potential target of failure. This paper extends this basic model to suit more realistic cases.

In reality, the arcs are not the only network elements that are at risk. The probability for a failure of substations in an electricity network, switching stations in a communication network, etc. should also be considered. These strategic points in the network might be of even higher importance to the network and its proper operation than the arcs, and the survival of them might be needed for a reliable service to its customers. When a node is unavailable, it is equivalent to a failure in all of the arcs that connect to this node, rendering them unavailable.

The application of security strategies is more often than not subject to a security budget that service providers have to their disposal. The goal of the service provider is to enable the service to all customers at all times. Previously, only connections between two points in the network, origin and destination of a service, have been considered. In a more realistic case, however, the connectivity between all nodes should be analysed. For example, in a communication network, every customer should be able to reach every other customer. In this paper, the model of **?** is extended to guide the service providers in their decision process of which security strategies to apply to maximise the probability of connectivity between all customers. This decision process is subjected to a security budget limitation.

To calculate the probability of a combination of failures to happen, and by extension the risk for any combination to disconnect service anywhere in the network, probability theory is used. Probability theory is used extensively in reliability theory and in reliability studies of systems, a field of research that has received a lot of attention in the past years. For an overview, the reader is referred to **???**.

The decision problem of selecting the best mix of security strategies given a budget limitation belongs to the more general category of knapsack problems. The knapsack problem represents a well-known class of combinational optimization problems (see **?** for more details). The model presented in this paper has a non-linear objective function and therefore belongs to the class of

non-linear knapsack problems, also known as non-linear resource allocation problems. The latter is known to be even more complex to solve than linear knapsack problems (?).

A subject that is closely related to the work in this paper, is that of survivable networks or network survivability (see ? for a survey). In this very active field of research, a network designer tries to increase or maximise the reliability of a network, however, the topology is not yet predefined. Two models have mainly been considered in this field. A model that is formulated as a minimum-cost network design problem with certain low-connectivity constraints (??) and the generalized Steiner problem which was first defined by ? and later renamed by ?. The network designer deals with an undirected graph, and tries to minimize the used budget selecting the optimal topology given constraints on minimal amount of flow between every node pair.

In ?, the reliability of stochastic-flow networks is evaluated. In the model described in that paper, both arcs and nodes can fail. The goal of their work is to evaluate the reliability of the network given a set of capacities for each arc and node, and a demand.

??, discuss the reliability of water distribution networks. The first technique to calculate the reliability, discussed in the first paper, is closely related to the technique discussed in the underlying paper. Our approach, however, is a more general implementation, as the first technique of the paper of ? only considers serial-parallel networks.

The problem of increasing security in a utility network can also be tackled using a game theoretical approach. More specifically, the attacker and the owner of the network can be treated as two opponents each one adopting different offensive or defensive strategies aimed at maximizing their own utility functions, which conflict each other. On the one hand, the goal of the owner of the network is to maximize the hypothetical benefits resulting from the customers of the network not being affected by service failures, considering that an investment in security is performed on certain network segments. On the other hand, a potential attacker focuses on maximizing the consequences of the attack considering its repercussions on the whole network. This scheme resembles the game theory model proposed in ? in which the problem of securing a transportation network used for dangerous goods is studied. The reader is referred to this work for more details about how to use a game theoretical approach to increase security in a transportation network. Nevertheless, the use of game theory is not within the scope of this paper.

3 Problem description

3.1 Arc and node failures

A utility network can be represented by a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{A}\}$, where $\mathcal{N}$ represents the set of nodes and $\mathcal{A}$ the set of arcs. All arcs $i \in \mathcal{A}$ and nodes $k \in \mathcal{N}$ have a probability of *failure*, denoted as p_i^a and p_k^n, where the index a refers to arcs and index n is used for the nodes. A node or arc failure always completely disables the respective node or arc.

A set of security strategies $\mathcal{S}_i^a$ and $\mathcal{S}_k^n$, is defined for each arc $i \in \mathcal{A}$ and each node $k \in \mathcal{N}$. For each security strategy $j \in \mathcal{S}_i^a$ (or $\mathcal{S}_k^n$) of arc i (node k) there is a cost c_{ij}^a (c_{kj}^n), and a value p_{ij}^a (p_{kj}^n), which is the probability of a failure of arc i (node k) when this security strategy is applied. A single security strategy needs to be selected for each node and each arc in the graph. The probability p_i^a of arc i failing and p_k^n of node k failing after applying a security strategy j will be equal to the probability p_{ij}^a or p_{kj}^n that is associated to that security strategy. In this paper, the assumption is made that a preliminary risk assessment phase has been conducted by experts, in order to determine the probability of failure associated with each arc or node, together with the costs and benefits of each available security measure.

As a side node, a security strategy can be a combination of several individual security measures (see e.g., Table 1). A combination of security measures can have a different effectiveness than the sum of the impact of the individual security measures due to some interaction effects. In some cases, combinations of single security measures might not be possible due to their incompatibility (e.g., in Table 1, Infra-red remote sensors and Thermal infra-red remote sensors are not compatible and hence this combination is not in the list of security strategies).

The default security strategy (labeled 0) for arc i, that has a cost $c_{i0}^a = 0$, is a base case that indicates that no security measure is applied. Its related probability p_{i0}^a represents the probability of a failure of arc i in case no security strategy is selected. This also applies to the security strategies and probabilities for the nodes.

Table 1: Examples of security strategies for pipelines

Strategy	Security measures	Cost	Probability
0	-	0	0.6
1	Fences	100	0.5
2	Infra-red remote sensors	150	0.45
3	Thermal infra-red remote sensors	200	0.4
4	Fences & Infra-red remote sensors	230	0.32
5	Fences & Thermal infra-red remote sensors	290	0.25

3.2 Mathematical formulation

The model defined in this paper selects a security strategy for each arc and each node such that the total risk of network failure is minimized, and a budget constraint is satisfied. In a previous paper (**?**), we have developed a simplified model with a given origin node o and a destination node d. In this model, a *network failure* occurs if these two nodes are disconnected, i.e., if no *functioning path* between o and d exists, where a functioning path is defined as a path that does not contain any failed nodes or edges. In case of a network failure, it is impossible for a service or good from node o to reach node d (e.g., it would be impossible to make a phone call from node o to node d). In this paper, we develop an extension of this basic model. The changes are twofold. Firstly, failures in nodes are also considered, as opposed to the simplified model,

where only arcs could fail. Secondly, a network failure is said to occur if any pair of nodes is disconnected, i.e., if two nodes exist in the graph such that no functioning path has these nodes as endpoints, while in the simple model only one origin and one destination node were considered.

The model developed in this paper minimizes the probability that a network failure will occur. To calculate the probability of network failure, a list of all *critical scenarios* is created. A scenario contains the state (failed or functioning) for each arc and each node. A *critical* scenario is a scenario that causes a network failure. We say that each element l of set C, the set of critical scenarios, is composed of arcs and nodes that fail (set $\mathcal{A}_l^E$ and $\mathcal{N}_l^E$ respectively), and arcs and nodes that do not fail (sets $\mathcal{A}_l^N$ and $\mathcal{N}_l^N$ respectively). It should be noted that $\mathcal{A}_l^E \cup \mathcal{A}_l^N \cup \mathcal{N}_l^E \cup \mathcal{N}_l^N = \mathcal{A} \cup \mathcal{N},\ \forall l \in C$. Given the failure probabilities for each arc and each node (which depend on the security strategies selected for these arcs and nodes), and given the state of each arc and node in scenario l, the occurrence probability of this scenario l, which we denote R_l, can be calculated as the product of the probabilities for each arc and node to be in their given state.

$$R_l = \prod_{i \in \mathcal{A}_l^E} p_i^a \cdot \prod_{i \in \mathcal{A}_l^N} (1 - p_i^a) \cdot \prod_{k \in \mathcal{N}_l^E} p_k^n \cdot \prod_{k \in \mathcal{N}_l^N} (1 - p_k^n) \tag{1}$$

Because scenarios are mutually exclusive events, the total probability of network failure (i.e., the objective function of the model developed in this paper), can be calculated by summing the probabilities of all critical events, i.e., $\sum_{l \in C} R_l$.

Obviously, the cardinality of C depends on the topology of $\mathcal{G}$, but will be very large in many situations. In this paper, we calculate the set C by checking all possible scenarios. Because each arc and each node can be in exactly two states (functioning or failed), the number of possible scenarios, each of which has to be checked for criticality, is $2^{|\mathcal{A}|+|\mathcal{N}|}$. Checking a scenario for criticality is equivalent to checking whether the graph in which the edges and nodes have been removed that fail in the scenario, is connected. In this paper, we use a simple breadth-first search algorithm.

In the model defined in this paper a budget constraint limits the security strategies that can be selected. Let B represent the available security budget and let x_{ij}^a be a binary variable that takes value 1 when security strategy j on arc i is applied, and 0 otherwise. Let x_{kj}^n be a binary variable that takes value 1 when security strategy j on node k is applied, and 0 otherwise. As mentioned, set $\mathcal{S}_i^a$ includes all the security strategies j for arc i with $j = 0$ being the situation in which no security measures for arc i are applied. The same applies for set $\mathcal{S}_k^n$ for all nodes $k \in \mathcal{N}$. A mathematical model to select the optimal security strategy for each arc and each node is the following.

$$\min \sum_{l \in C} R_l \tag{2}$$

s.t.

$$\sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{S}_i^a} c_{ij}^a \cdot x_{ij}^a + \sum_{k \in \mathcal{N}} \sum_{j \in \mathcal{S}_k^n} c_{kj}^n \cdot x_{kj}^n \leq B \tag{3}$$

$$p_i^a = \sum_{j \in \mathcal{S}_i^a} p_{ij}^a \cdot x_{ij}^a \qquad \forall i \in \mathcal{A} \tag{4}$$

$$p_k^n = \sum_{j \in \mathcal{S}_k^n} p_{kj}^n \cdot x_{kj}^n \qquad \forall k \in \mathcal{N} \tag{5}$$

$$R_l = \prod_{i \in \mathcal{A}_l^E} p_i^a \cdot \prod_{i \in \mathcal{A}_l^N} (1 - p_i^a) \cdot \prod_{k \in \mathcal{N}_l^E} p_k^n \cdot \prod_{k \in \mathcal{N}_l^N} (1 - p_k^n) \qquad \forall l \in C \tag{6}$$

$$\sum_{j \in \mathcal{S}_i^a} x_{ij}^a = 1 \qquad \forall i \in \mathcal{A} \tag{7}$$

$$\sum_{j \in \mathcal{S}_k^n} x_{kj}^n = 1 \qquad \forall k \in \mathcal{N} \tag{8}$$

$$x_{ij}^a \in \{0, 1\} \qquad \forall i \in \mathcal{A}, \forall j \in \mathcal{S}_i^a \tag{9}$$

$$x_{kj}^n \in \{0, 1\} \qquad \forall k \in \mathcal{N}, \forall j \in \mathcal{S}_k^n \tag{10}$$

The objective function in Eq. (2) minimizes the total probability of network failure, and is calculated as the sum of probabilities of all critical scenarios. Constraint Eq. (3) ensures that the total cost associated to the selected security strategies does not exceed the predefined security budget B. Equations (4) and (5) are used to determine the probability p_i^a of failure of arc i and the probability p_k^n of a failure of node k. Equation (6) calculates the occurrence probabilities of all critical scenarios. Equations (7) and (8) force the decision process the selection of exactly one security strategy for each arc or node, where $x_{i0}^a = 1$ or $x_{k0}^n = 1$ indicate that no security strategy (or security strategy 0) has been selected for arc i and node k respectively. Finally, Eqs. (9) and (10) enforce the domain of the decision variables, and ensure that no partial security strategies are allowed.

4 Solution approach

4.1 Auxiliary calculations

The model presented in the previous section is computationally expensive because of the number of scenarios, and the resulting risk calculations. If the security strategy of a single node or arc changes, the failure probability changes and the occurrence probability of *every* critical scenario is affected. This can be seen in Eq. (6), in which the failure probabilities of each arc and node appear. Fortunately, it is not necessary to recalculate Eq. (6) from scratch every time the security strategy of an arc or node changes.

The implementation of the proposed algorithm uses an update formula to recalculate the risk for each scenario when a security strategy is changed.

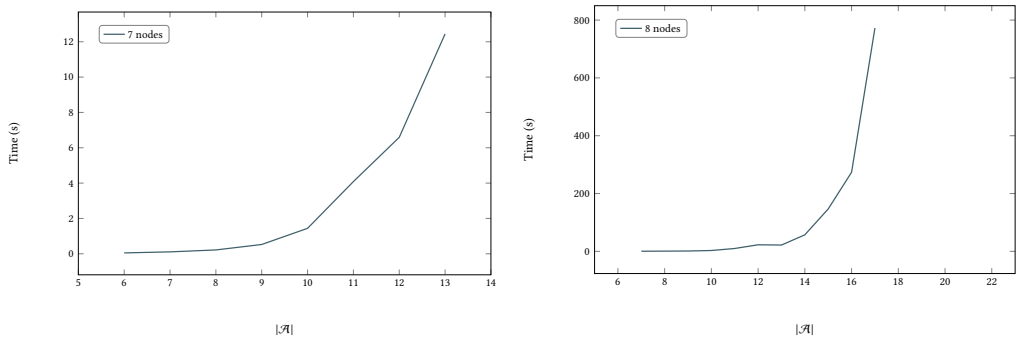
$$\bar{R}_l = \frac{R_l \cdot \bar{P}}{P} \tag{11}$$

The formula divides the previous probability for the scenario, R_l, with the previous failure probability of the arc or node P, and multiplies it with the new probability, $\bar{P}$, to obtain the new probability of the scenario, $\bar{R}_l$. If the arc or node is failing in that scenario, P is substituted with p_x, and $(1 - p_x)$ otherwise. This only requires two operations per scenario (one multiplication, one division), whereas recalculating Eq. (6) from scratch results in $|\mathcal{A}| + |\mathcal{N}|$ calculations per scenario.

4.2 Exact approach

To establish a benchmark for the heuristic developed in Section 4.3, we have implemented a naive exact approach that loops over each combination of security strategies and validates them for the budget constraint. If the combination of security strategies does not infringe the budget constraint, the exact approach then verifies if it has a better result than the previous best found solution, and stores it if it has.

Exact approaches to solve the problem, described in Eqs. (2) to (10), are viable methods for small instances only, due to the exponential explosion of the number of critical combinations to be considered. In fact, in the worst case, the number of critical combinations that have to be analysed and updated through the solution process is equal to $2^{|\mathcal{A}|+|\mathcal{N}|}$. The maximum number of possible ways to combine security strategies is equal to $\prod_{i \in \mathcal{A}} |\mathcal{S}_i^a| \cdot \prod_{k \in \mathcal{N}} |\mathcal{S}_k^n|$. In case the number of security strategies is equal for all nodes and arcs and simplified as $|\mathcal{S}|$ this value can be rewritten as $|\mathcal{S}|^{|\mathcal{A}|+|\mathcal{N}|}$. As an illustration of the combinatorial explosion that results, Fig. 1 shows the influence of the number of arcs on the computational time for instances of 7 and 8 nodes.



(a) Number of arcs versus CPU time for an instance with 7 nodes

(b) Number of arcs versus CPU time for an instance with 8 nodes

Figure 1: Relationship between the number of arcs and the CPU time needed for the exact algorithm

4.3 Heuristic approach

Given the computational complexity of the problem, it is unlikely that an efficient exact approach can be developed. Therefore, we develop an efficient metaheuristic approach, sacrificing a guarantee of optimality in favor of finding near-optimal solutions in short running times.

Our algorithm is an iterated local search algorithm (ILS) (**?**) which is hybridised with a greedy random adaptive search procedure (GRASP) (**???**) and a variable neighbourhood descent (VND) improvement heuristic. Two perturbation heuristics are used to escape from local optima. In addition, a tabu list is used during the whole execution of the heuristic to avoid an exploration of solutions that have been analysed in previous iterations. Pseudo-code of the algorithm can be found in Algorithm 1.

Algorithm 1 Metaheuristic structure

Step 0: Initialization
Read instance & Initialize heuristic parameters:
`iter` $\leftarrow 0$ # number of iterations without improvement
`loopcount` $\leftarrow 0$ # number of restarts
$x^* \leftarrow \emptyset$, $f(x^*) \leftarrow \infty$
Step 1: Construction phase
$x \leftarrow$ `GRASP`
while `loopcount` $<$ `maxloopcount` **do**
  Step 2: Intensification phase
  $k \leftarrow 0$
  while $(k < k_{\max})$ **do**
    $x' \leftarrow$ `VND`$(N_k(x))$ # find the best solution in the kth neighborhood
    if $(f(x') < f(x))$ **then**
      $x \leftarrow x', f(x) \leftarrow f(x')$ # if a better solution is found, search in the same neighborhood
    else
      $k \leftarrow k + 1$ # otherwise go to the next neighborhood
    end if
  end while
  if $(f(x) < f(x^*))$ **then**
    $x^* \leftarrow x, f(x^*) \leftarrow f(x)$ # if the best solution is improved, update it
  else
    `iter` $\leftarrow$ `iter` $+ 1$ # otherwise, update number of iterations without improvement
  end if
  Step 3: Diversification phase
  if `iter` $<$ `maxiter` **then** # number of iterations without improvement not reached
    $x \leftarrow$ `Perturbation`(x)
  else
    $x \leftarrow$ `GRASP`
    `loopcount` $\leftarrow$ `loopcount` $+ 1$
  end if
end while
return x^*

An initialisation step is performed, followed by a construction phase, in which a GRASP

heuristic is used to construct an initial solution. The solution found is then passed to a variable neighborhood descent (VND), which performs different moves on the solution, until no more improvement can be found, after which a differentiation phase is executed.

After the initialization step, an initial solution is generated by a GRASP constructive heuristic. The GRASP heuristic starts from a solution in which all arcs and nodes have the default security strategy (strategy 0). The total cost of such a solution is zero. One step at a time, the GRASP heuristics upgrades security strategies for arcs and nodes. The GRASP heuristic repeats the selection of an arc or node and an upgraded security strategy for that arc or node until the security budget does not allow any further security strategy upgrades. The selection of the arc or node to upgrade the security strategy for, happens by selecting a random arc or node from a restricted candidate list (RCL).

A first and best improvement strategy are implemented in the GRASP heuristic, and based on a parameter either one of them is used. For the best improvement strategy, all possible security upgrades for the selected arc or node are evaluated, and a random security strategy of the most improving security strategies is selected and applied as upgrade to its arc or node. For the first improving strategy, the security strategy is selected at random from the whole list of security strategies for the selected arc or node. If, at the end of the algorithm, for the first improvement strategy the randomly selected security strategy is out of budget, a other security strategy is selected at random for that arc or node, and evaluated for the budget, until a solution strategy can be used as an upgrade, or all security strategies for that arc or node are evaluated.

The RCL is created by sorting all nodes and arcs by their "potential" to reduce the probability of network failure, and selecting the top `alpha` arcs/nodes from that list. This potential is based on the frequency of occurrence of arcs and nodes in critical scenarios, the probability of occurence of that scenario, and the amount of disconnection it causes in the network. If an arc or node is part of many critical scenarios (in a failed state), it is likely to have a higher potential to improve the network failure probability when their own failure probability is decreased. Arcs or nodes that (in a failed state) do not appear in many critical scenarios, are similarly unlikely to improve the network failure probability.

The calculation of the potential for each arc and node is done by a method that ranks the set of critical combinations, C, based on an index of connectivity for each scenario Con_l. Given a scenario l, this index of connectivity can be calculated as the fraction of node pairs between which a functional path in the graph exists. Given a graph with n nodes, where $n = |\mathcal{N}|$ the index of connectivity Con_l, is calculated as follows:

$$Con_l = \frac{\sum_{o=1}^{n} \sum_{d=1}^{n} r_{od}^{l}}{n \cdot (n-1)}, \tag{12}$$

where r_{od}^{l} is 1 if there is a functional path from node o to node d under scenario l, and 0 otherwise.

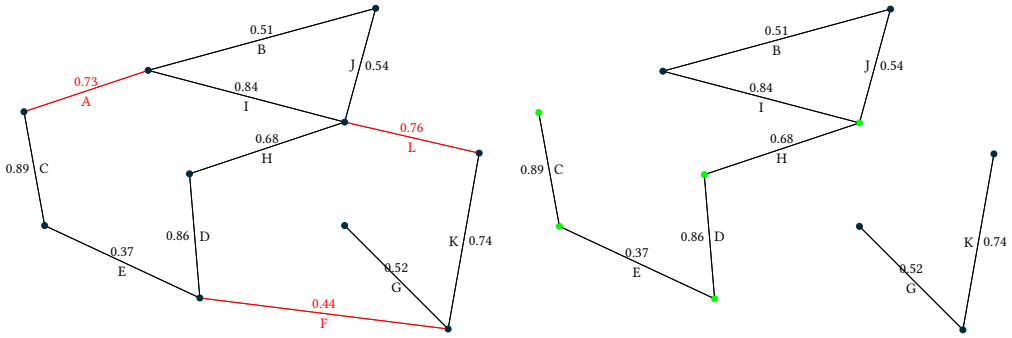
To find $\sum_{o=1}^{n} \sum_{d=1}^{n} r_{od}^{l}$, a simple breadth-first search algorithm is used, which executes the following steps.

Step 1: Starting from an initial node, all adjacent nodes (connected by an edge that is not failed) are assigned to the same group. Those neighbors become the new initial nodes, and their adjacent nodes are assigned to the same group (see Fig. 2(b)). If no more neighbors are found, go to step 2.

Step 2: The algorithm moves to the next node that has not been assigned to a group yet and applies the same steps (see **??**). If no more nodes are left unassigned, the algorithm is finished (see **??**), otherwise, repeat step 2.

To calculate the number of disconnected node pairs, the cardinality of each group is multiplied with that of every other group, and then summed.

Finally, to find Con_l, the fraction of disconnected node pairs, $\sum_{o=1}^{n} \sum_{d=1}^{n} r_{od}^{l}$, is divided with the total number of theoretical connections to get a percentage value, which is then subtracted from 1 to get the percentage of connectivity. In Fig. 2 and **??**, this would be one group of 7 and one group of 3. The total number of connections possible is 45. The rate of disconnection is 47.7%. It is important to take in to account that in our examples the connection from A to B is the same as the connection from B to A, as the graph is undirected. In case of a directed graph this formula and algorithm has to be adapted to be suitable to calculate the connectivity index.



(a) Initial graph with 3 failures

(b) Algorithm marking first group of neighbors that are connected

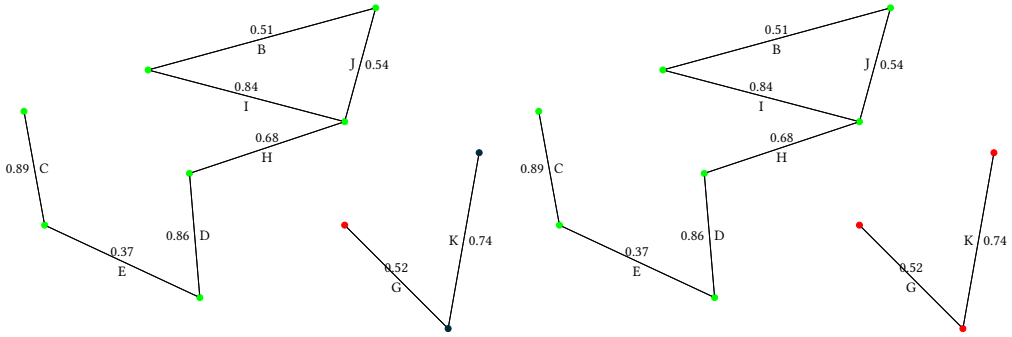
Figure 2:

Using the probability for critical scenario l to occur (R_l), ρ_i^a and ρ_k^n are defined as the *potential* of arc i and node k as follows:

$$\rho_i^a = \sum_l (1 - Con_l) \cdot R_l \cdot f_{il}^a \tag{13}$$

$$\rho_k^n = \sum_l (1 - Con_l) \cdot R_l \cdot f_{kl}^n \tag{14}$$

where f_{il}^a (f_{kl}^n) is 1 if the arc i (node k) is present in set $\mathcal{A}_l^E$ ($\mathcal{N}_l^E$), 0 otherwise.



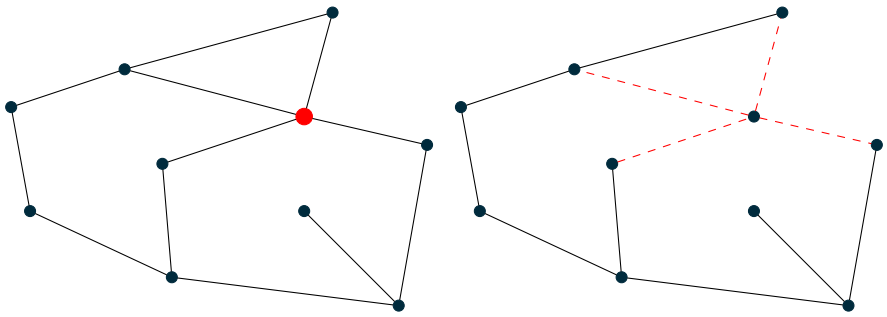
(a) Algorithm starts marking second group (b) No more unassigned nodes left

Figure 3: Marking disconnected groups for calculating the connectivity index

The potential of arc i and node k is a measure of the importance of that arc or node in determining the probability of network failure, and hence also of the likelihood for an upgrade of the security strategy of that arc or node to improve this probability. It uses a combination of the risk for a critical scenario to occur, weighted with the impact of this scenario (which is one minus the amount of connections left if a critical scenario takes place), and this value is then summed for all the scenarios in which that arc or node fail.

This potential is used to generate the restricted candidate list (RCL) for the GRASP heuristic, as well as, for finding promising arcs and nodes for the VND. The arcs and nodes are all put together in a list and sorted by their potential. The top `alpha` elements are then selected to form the RCL.

When a node fails, this is equivalent to the failure of all the incident arcs. In other words, if a node is not available due to a failure, the arcs entering in and leaving from that node need to be considered out of service. In the calculations of the index of connectivity a failure of a node is treated as the failure of the arcs that enter or leave from the node. An example is shown in **??**



(a) Attack on a node (b) Equivalent failure of arcs

Figure 4: Substitution of a failure of a node (**??**) with the failure of arcs (**??**) for the computation of the index of connectivity

After the generation of an initial solution by the GRASP heuristic, a local search heuristic is applied during the Intensification step. The local search makes use of a VND heuristic to improve the current solution. The VND heuristic explores the neighbourhoods of three different local search operators. The first one "*Upgrade*" tries to upgrade one security strategy used inside an arc (or node) with another security strategy for said arc (or node). The second move, "*Budget reallocation*", tries to relocate budget from one arc to another by downgrading a security strategy used for one arc (or node) and applying the budget to antoher arc (or node) by upgrading a security strategy on that arc (or node). The third move, "*Budget redistribution*", downgrades the security strategies from two arcs (or nodes) and they are then afterwards upgraded with different security strategies for the same two nodes or arcs (this can be seen as a budget redistribution between the selected arcs or nodes). A move is executed until no further improvement is found. The whole VND heuristic is carried out until a local optimum is found and no further improvement can be obtained.

Finally, a perturbation is applied to escape this local optimum (this is a diversification step), and the algorithm continues with a local search on this perturbed solution. In this perturbation step, part of the solution is destroyed by removing security strategies from arcs or nodes randomly. These security strategies are added to the tabu list for a specified number of iterations. The arcs or nodes they belonged to are also added to a tabu list for a predefined number of iterations. If, after a fixed number of perturbations, the algorithm cannot find a better solution, the algorithm is restarted from a new solution constructed by the GRASP heuristic. The perturbation step in this algorithm does not require a repair step, as a solution from which a security strategy is removed is always feasible.

5 Computational results

The solution approach, described before, has been tested on a set of realistic instances, which are available on `http://antor.uantwerpen.be/downloads/NS`. The instances are randomly generated, taking into account realistic features of networks and failure rates. The values of the parameters that were passed to the instance generator to generate a specific instance are encoded in the instance name.

The generated nodes that are positioned randomly in a uniform fashion are connected by a minimal spanning tree, and in a next step, edges are randomly selected from the Delaunay triangulation of those nodes to be added to the instance. The maximum number of edges that can be added is defined in the input parameters. All the nodes and edges are assigned a probability of failure which is randomly selected, and which is of a realistic value. From **?** we can learn that, for a real 16 year old water network the failure rate is between 0.5% and 1.5%. Some preliminary tests reveal that the exact failure rate does not make a difference for the effectiveness and speed of the heuristic approach proposed in this paper. Next each edge and node are assigned between 1 and `c` countermeasures (`c` is a user-defined parameter), which are combined in security strategies for that edge or node. Each security strategy has a reduction of risk and a cost.

The encoding of the instances' name is as follows: `NS-n8-c5-C3-a30-x0`, where `n` represents the number of nodes in the instance which are generated at a random position, `c` gives the maximum number of counter measures generated for each edge, `C` is the maximum number of connections from and to a node (This is currently not used in the instance generator, but the option is foreseen in the encoding), `a` represents the percentage of extra arcs from the Delaunay triangulation added to the minimal spanning tree, and `x` represents the number of the instance when multiple instances are generated at once, with the same settings.

All computational experiments are conducted on a desktop running a 64 bit version of Linux. The system has a Intel® Core™ i7-4790 processor, running at 3.60GHz, and has 16GB of memory.

5.1 Calibration of the metaheuristic

In a first phase, the metaheuristic has been tuned in order to find its best parameter settings. This tuning is done in a controlled full factorial statistical experiment on a subset of the instances. The parameters that were investigated are the maximum number of iterations (`maxiter-no-improvement`), the percentage of the solution that is perturbed (`perturb-percent`), the size of the tabu list (`tabu-tenure`), the selection mechanism (`selection-mechanism`), the size of the restricted candidate list (`alpha`), and the percentage of the instance used for the third move (`double-swap-percentage`, see Section 4).

Analysis of variance (ANOVA) reveals that the percentage of the solution that is perturbed, the size of the tabu list, the size of the restricted candidate list, and the percentage of the instance used for the third move all have a statistically significant influence on the objective function value. A graphical output of the statistical experiment is shown in **??**. From the full factorial experiment we choose the parameter settings shown in **??**.

Table 2: Heuristic parameters

Parameter	Description	Values	Chosen value
`max-iter`	Number of restarts	50	50
`perturb-percent`*	Percent of edges removed during the perturbation phase	10%, 30%, 50%, 70%	70%
`alpha`*	Size of the restricted candidate list	1, 2, 3, 4	3
`tabu-tenure`*	Number of iterations that a strategy is kept in the tabu list	10, 30, 50	30
`max-iter-no-improvement`	Number of iterations without improvements	5, 10, 20	10
`double-swap-percentage`*	Percentage value used to find the amount of evaluations to be performed by the Double swap move	20%, 50%, 80%	80%
`selection-mechanism`	Strategy to select edges	best (0), first(1)	best

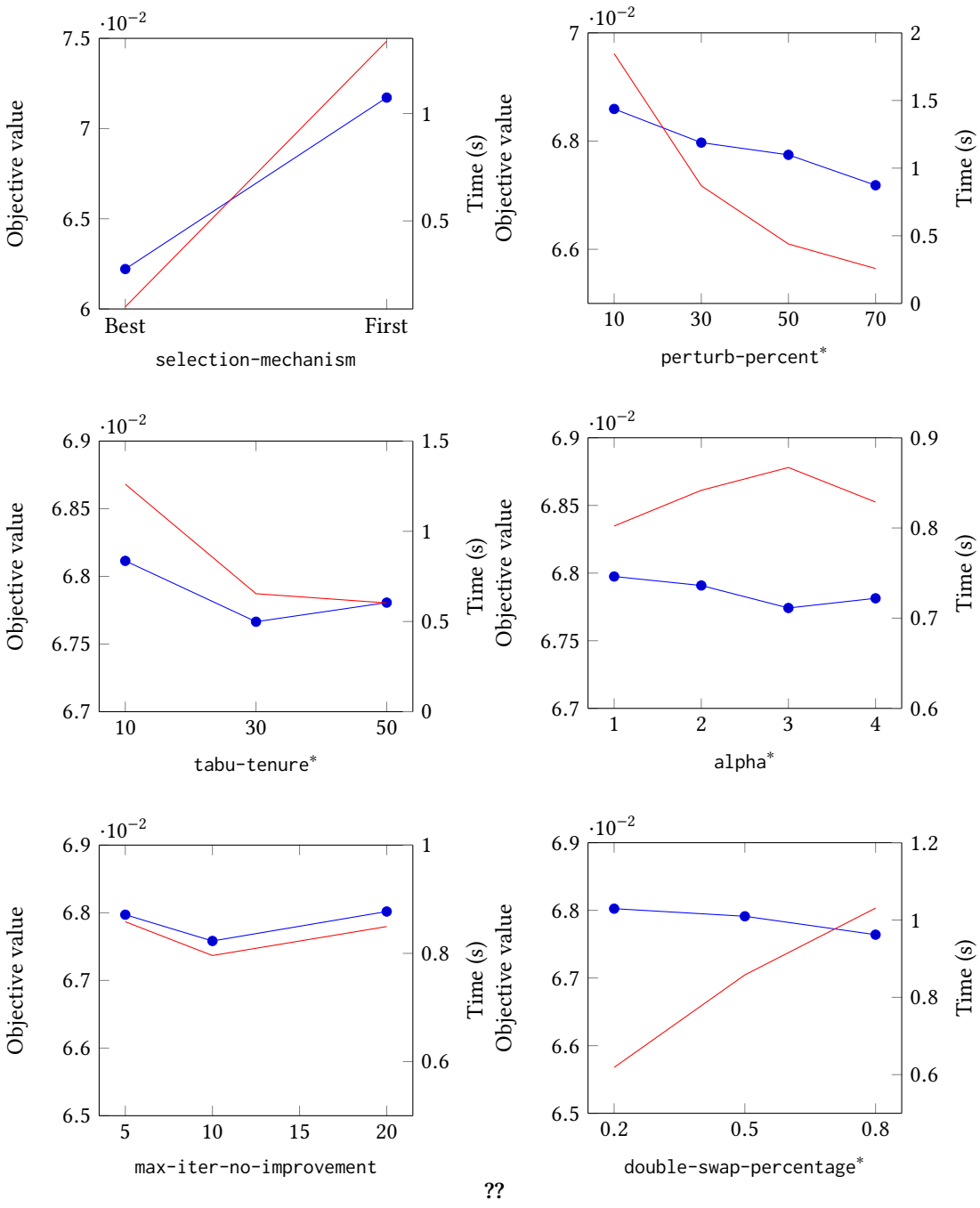


Figure 5: Plot of average objective values and computing times for given parameter setting

5.2 Results

In a second phase, the algorithm is executed with these best parameter settings on a larger set of instances to analyse its behavior. The obtained results are compared with the best solutions found by the naive exact approach described earlier. If a solution was not found within 24 hours by the exact approach, no value is given.

In **??**, the percentage difference between the best solutions obtained by the metaheuristic over 25 runs and the exact approach is reported in column *Best Gap*, while column *Average Gap* represents the percentage gap between the average solution over 25 runs and the best solution found by the exact approach. It can be observed that the metaheuristic finds, in most cases, solutions that are equal to those found by the exact approach. Moreover, these solutions of high quality are also found in a very limited running time, i.e., a small fraction of the time used by the exact approach.

In **??**, the current and best found objective values for one instance are plotted over time. The spikes in the current solution show the points in time at which a perturbation is executed. After a perturbation, the VND heuristic lowers the objective value in small steps until it reaches a local optimum, after which another perturbation is executed. The objective function has a very steep descent at the start of the algorithm and gradually stabilizes. This shows that the algorithm converges towards very good results in a very short time. As CPU time gets larger, the marginal improvement of the best solution found so far becomes smaller and smaller. This shows that infinitely increasing the CPU time will not necessarily produce significantly better results.

6 Conclusion and discussion

In this paper, a model for the selection of appropriate security strategies given a limited budget is described. Its goal is to increase the security of infrastructure such as pipelines transportation systems, telecommunication networks, smart grids, etc.

Very good results are obtained by the proposed heuristic on the provided instances. In most cases, it finds the optimal solution in a fraction of the time that is needed by the exact approach. The parameters that have a significant influence on the results are the perturbation percentage, where a lower percentage yielded better results, the tabu tenure, the number of elements in the restricted candidate list, alpha, and the percentage of arcs and nodes on which the budget redistribution move is executed.

However, due to the fact that we need to generate and evaluate all scenarios to find the critical scenarios and the sheer number of critical scenarios, to tackle larger instances, a different approach should be investigated.

In future research, the model can be extended even more by making a differentiation in the types of nodes (customers or suppliers) and/or considering the importance of nodes themselves. In terms of the algorithm, alternative solution approaches can also be developed and compared

Table 3: Exact approach in comparison with the metaheuristic

Instance	Exact Approach		Metaheuristic				
	Best Solution	Time (s)	Best Solution	Avg. Solution	Avg. Time (s)	Best Gap (25 runs)	Avg. Gap (25 runs)
NS-n5-c5-C3-a30-x0	0.064	134	0.064	0.070	0.008	0.000%	9.616%
NS-n5-c5-C3-a30-x1	0.051	502	0.051	0.057	0.011	0.609%	10.409%
NS-n5-c5-C3-a30-x10	0.057	673	0.057	0.062	0.009	0.000%	9.034%
NS-n5-c5-C3-a30-x11	0.064	218	0.064	0.070	0.008	0.375%	9.457%
NS-n5-c5-C3-a30-x12	0.060	70	0.060	0.065	0.007	0.000%	8.768%
NS-n5-c5-C3-a30-x13	0.065	147	0.065	0.071	0.008	0.009%	9.060%
NS-n5-c5-C3-a30-x14	0.059	391	0.060	0.065	0.008	0.521%	10.135%
NS-n5-c5-C3-a30-x2	0.046	759	0.046	0.047	0.028	0.000%	2.399%
NS-n5-c5-C3-a30-x3	0.062	353	0.062	0.067	0.009	0.000%	8.241%
NS-n5-c5-C3-a30-x4	0.062	447	0.062	0.069	0.009	0.000%	9.723%
NS-n5-c5-C3-a30-x5	0.055	1229	0.055	0.057	0.033	0.000%	3.558%
NS-n5-c5-C3-a30-x6	0.045	3785	0.045	0.046	0.034	0.000%	1.149%
NS-n5-c5-C3-a30-x7	0.063	86	0.063	0.069	0.010	0.000%	10.047%
NS-n5-c5-C3-a30-x8	0.051	48	0.051	0.056	0.006	0.000%	10.052%
NS-n5-c5-C3-a30-x9	0.056	453	0.056	0.060	0.009	0.000%	7.388%
NS-n6-c5-C3-a30-x0	0.062	88004	0.062	0.063	0.192	0.005%	1.433%
NS-n6-c5-C3-a30-x1	0.054	37374	0.054	0.056	0.125	0.000%	2.847%
NS-n6-c5-C3-a30-x10			0.073	0.075	0.223		
NS-n6-c5-C3-a30-x11	0.056	26487	0.056	0.058	0.170	0.000%	2.649%
NS-n6-c5-C3-a30-x12			0.058	0.060	0.223		
NS-n6-c5-C3-a30-x13	0.074	45078	0.074	0.077	0.186	0.000%	3.445%
NS-n6-c5-C3-a30-x14	0.082	25911	0.082	0.084	0.155	0.000%	1.751%
NS-n6-c5-C3-a30-x2	0.067	68352	0.067	0.068	0.200	0.000%	1.534%
NS-n6-c5-C3-a30-x3	0.066	58443	0.066	0.067	0.250	0.000%	2.828%
NS-n6-c5-C3-a30-x4	0.062	13316	0.062	0.064	0.165	0.000%	2.405%
NS-n6-c5-C3-a30-x5	0.076	72039	0.076	0.077	0.112	0.000%	2.031%
NS-n6-c5-C3-a30-x6	0.063	9058	0.063	0.064	0.169	0.000%	1.560%
NS-n6-c5-C3-a30-x7			0.064	0.065	0.300		
NS-n6-c5-C3-a30-x8			0.063	0.063	0.316		
NS-n6-c5-C3-a30-x9			0.058	0.059	0.216		
NS-n7-c5-C3-a30-x0			0.097	0.098	5.069		
NS-n7-c5-C3-a30-x1			0.075	0.077	5.082		
NS-n7-c5-C3-a30-x10			0.090	0.091	4.991		
NS-n7-c5-C3-a30-x11			0.083	0.085	2.856		
NS-n7-c5-C3-a30-x12			0.050	0.052	2.566		
NS-n7-c5-C3-a30-x13			0.090	0.091	3.286		
NS-n7-c5-C3-a30-x14			0.081	0.083	3.453		
NS-n7-c5-C3-a30-x2			0.068	0.070	4.550		
NS-n7-c5-C3-a30-x3			0.085	0.087	4.225		
NS-n7-c5-C3-a30-x4			0.097	0.098	6.083		
NS-n7-c5-C3-a30-x5			0.084	0.085	4.897		
NS-n7-c5-C3-a30-x6			0.085	0.086	3.049		
NS-n7-c5-C3-a30-x7			0.091	0.093	2.690		
NS-n7-c5-C3-a30-x8			0.083	0.084	4.173		
NS-n7-c5-C3-a30-x9			0.078	0.079	4.591		

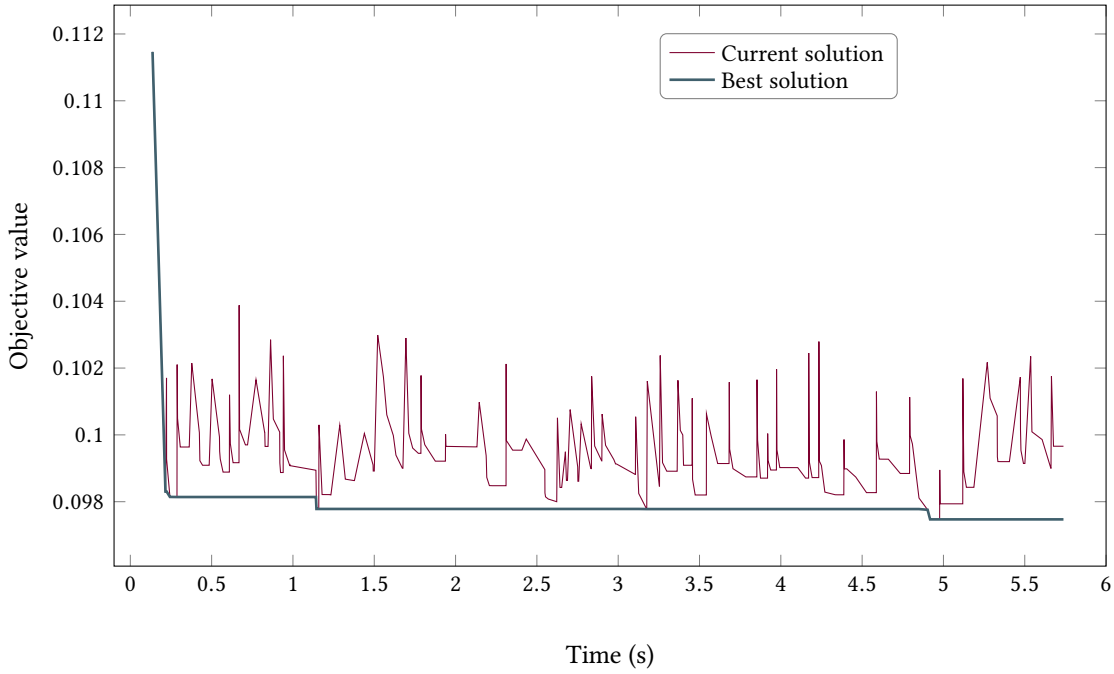


Figure 6: Plot of the objective value over time.

with the one proposed in this paper. It might also be worth investigating if a faster approach can be found to calculate the set C, without explicitly calculating and enumerating all possible combinations. A very good implementation of an exact approach should be considered, to find the optimal solution for a set of test instances, in a reasonable amount of time.

Acknowledgements

This research is supported by the Interuniversity Attraction Poles (IAP) Programme initiated by the Belgian Science Policy Office (COMEX Project).

References

I. Bazovsky. *Reliability Theory and Practice*. Dover Civil and Mechanical Engineering Series. Dover Publications, 2004.

K. M. Bretthauer and B. Shetty. The nonlinear knapsack problem – algorithms and applications. *European Journal of Operational Research*, 138(3):459 – 472, 2002.

M. Bruch, V. Münch, M. Aichinger, M. Kuhn, M. Weymann, and G. Schmid. Power blackout risks. In *CRO Forum*, 2011.

M. Fadaee and R. Tabatabaei. Estimation of failure probability in water pipes network using statistical model. *World Applied Sciences Journal*, 11(9):1157–1163, 2010.

P. Festa and M. G. Resende. Grasp: An annotated bibliography. In *Essays and surveys in metaheuristics*, pages 325–367. Springer, 2002.

P. Festa and M. G. Resende. An annotated bibliography of grasp–part i: Algorithms. *International Transactions in Operational Research*, 16(1):1–24, 2009a.

P. Festa and M. G. Resende. An annotated bibliography of grasp–part ii: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009b.

M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, 40(2): 309–330, 1992a.

M. Grötschel, C. L. Monma, and M. Stoer. Facets for polyhedra arising in the design of communication networks with low-connectivity constraints. *SIAM Journal on Optimization*, 2(3):474–504, 1992b.

J. Janssens, L. Talarico, and K. Sörensen. A hybridised variable neighborhood tabu search heuristic to increase security in a utility network. *Reliability Engineering & System Safety*, 2015. doi: http://dx.doi.org/10.1016/j.ress.2015.08.008.

H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.

Y.-K. Lin. A simple algorithm for reliability evaluation of a stochastic-flow network with node failure. *Computers & Operations Research*, 28(13):1277–1285, 2001.

H. Lourenço, O. Martin, and T. Stützle. *Iterated Local Search: Framework and Applications*. Springer New York, 2010.

Ministery of Defence (UK). Chapter 6: Probabilistic R&M Parameters and redundancy calculations. In *Applied R&M Manual for Defence Systems (GR-77), Part D - Supporting Theory*. UK Ministery of Defence, Abbey Wood, Bristol, 2011.

L. P. Rees, J. K. Deane, T. R. Rakes, and W. H. Baker. Decision support for cybersecurity risk planning. *Decision Support Systems*, 51(3):493 – 505, 2011.

G. Reniers and W. Dullaert. Tepitri: A screening method for assessing terrorist-related pipeline transport risks. *Security Journal*, 25(2):173–186, 2012.

G. Reniers, K. Sörensen, and W. Dullaert. A multi-attribute systemic risk index for comparing and prioritizing chemical industrial areas. *Reliability Engineering & System Safety*, 98(1):35–42, 2012.

J. L. Romeu. Understanding series and parallel systems reliability. *Selected Topics in Assurance Related Technologies (START)*, 1(5):1 – 8, 2004.

T. Sawik. Selection of optimal countermeasure portfolio in {IT} security planning. *Decision Support Systems*, 55(1):156 – 164, 2013.

K. Steiglitz, P. Weiner, and D. Kleitman. The design of minimum-cost survivable networks. *Circuit Theory, IEEE Transactions on*, 16(4):455–460, 1969.

L. Talarico, G. Reniers, K. Sörensen, and J. Springael. Mistral: A game-theoretical model to allocate security measures in a multi-modal chemical transportation network with adaptive adversaries. *Reliability Engineering & System Safety*, 138:105–114, 2015.

J. M. Wagner, U. Shamir, and D. H. Marks. Water distribution reliability: analytical methods. *Journal of Water Resources Planning and Management*, 114(3):253–275, 1988a.

J. M. Wagner, U. Shamir, and D. H. Marks. Water distribution reliability: simulation methods. *Journal of Water Resources Planning and Management*, 114(3):276–294, 1988b.

C. Wilbaut, S. Hanafi, and S. Salhi. A survey of effective heuristics and their application to a variety of knapsack problems. *IMA Journal of Management Mathematics*, 19(3):227–244, 2008.

P. Winter. Steiner problem in networks: a survey. *Networks*, 17(2):129–167, 1987.