DEPARTMENT OF ENGINEERING MANAGEMENT

A hybridised tabu search heuristic to increase security in a utility network

Jochen Janssens, Luca Talarico & Kenneth Sörensen

UNIVERSITY OF ANTWERP
Faculty of Applied Economics

City Campus
Prinsstraat 13, B.226
B-2000 Antwerp
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
www.uantwerpen.be

FACULTY OF APPLIED ECONOMICS

DEPARTMENT OF ENGINEERING MANAGEMENT

A hybridised tabu search heuristic to increase security in a utility network

Jochen Janssens, Luca Talarico & Kenneth Sörensen

RESEARCH PAPER 2014-023
OCTOBER 2014

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.226
phone: (32) 3 265 40 32
fax: (32) 3 265 47 99
e-mail: joeri.nys@uantwerpen.be

D/2014/1169/023

A hybridised tabu search heuristic to increase security in a utility network

Jochen Janssens, Luca Talarico, Kenneth Sörensen

University of Antwerp Operations Research Group ANT/OR

Prinsstraat 13, 2000 Antwerp, Belgium

October 2014

Abstract

In this paper we propose a decision model aimed at increasing security in a utility network (e.g., smart grid, water network). The network consists of edges that might be a viable target for terrorists depending on several factors such as geographical location, accessibility of the network, the political stability of a region. Based on this we assume that all edges (e.g., pipes, cables) have a certain, not necessary equal, probability of failure, which can be reduced by selecting appropriate edge-specific security strategies. The decision model proposed in this paper is based on a metaheuristic approach, that uses tabu search hybridised with iterated local search and a large scale neighbourhood decent heuristic. The main goal is to reduce the risk of service failure between a couple of network nodes by selecting the right combination of security measures for each network edge given a limited security budget. A generator for realistic instances is proposed and used to create a set of test instances. Experiments on these instances are conducted to tune the parameters and evaluate the proposed metaheuristic algorithm.

Keywords: network security, metaheuristics, knapsack problem.

1 Introduction

In modern day society, utility networks such as electricity, water, gas, and communication networks are taken for granted. People expect that they function at all times, and are capable of handling all demand placed on them. However, there is a real risk of failures in all types of networks. Those failures might make the network unavailable with a resulting interruption of the service/connection between two network points which are

represented by an origin node (i.e., the point from which the service or the product is sent to the customer through the network) and a destination node (i.e., the customer or the point to which the product, service is delivered through the network).

Network breakdowns can have *safety*-related causes such as natural phenomena (e.g., earthquakes, storms), human errors, or mechanical defects such as in pumps and valves, caused by the regular wear and tear. In addition, network breakdowns can be due to *security*-related causes such as intentional terrorist attacks and/or malicious sabotages. We refer the reader to Reniers et al. (2008) for a further clarification of terms *safety* and *security*.

After 9/11, the protection of utility networks against intentional attacks has received great attention among network providers. In fact, terrorist attacks on utility network are not rare and might cause huge losses in a nation's economy (START: A Center of Excellence of the U.S. Department of Homeland Security, 2014). For these reasons in the remainder of the paper our focus is on network security rather than safety.

Network providers and managers can reduce the risk of a network breakdown after a terrorist attack addressed at one or several network edges by applying preventive security measures in order to reduce network vulnerabilities.

The security budget that can be spent on these security measures, however, is generally limited. The current economic situation has increased the pressure on limiting even further many budgets and investments in security. In this work, a combination of security measures for one edge is called a security strategy for that edge.

The problem defined in this paper is, how to allocate a budget between edge-specific security strategies in order to reduce the risk of a utility network being (partially) out of service. This risk is measured as the probability that a failure would occur, and the fact if it would break the connection between an origin and a destination node.

Since the budget is limited and the security strategies can only be applied locally, i.e., on a specific link in the network, the security strategies should be chosen in such a way that the reduction of the risk of the network service being down is as large as possible while keeping the total cost of the security strategies within the budget.

Once we consider realistic cases with a larger number of edges and different security strategies (in this paper from 5 up to 20), the problem can turn out to be so large that it will become computationally infeasible to solve it in a reasonable amount of time with exact algorithms. Therefore we will explore the use of metaheuristics to support this decision problem.

The paper is organized as follows. In Section 2, we give a brief overview of the state of the art. Section 3 clarifies the problem of selecting the best strategies to increase the security of the whole network. It is described and modelled as an optimisation problem. An illustrative example is also provided. In Section 4, we present a metaheuristic to solve the network security problem. Section 5.1 presents the instance generator, reports

the parameter tuning step and the results of the computational experiments. Section 6 concludes the paper and presents some ideas for further developments.

2 Literature review

Following the 9/11 attacks, network security has received growing attention within the scientific community. Although significant research has been done to improve best practices in the field of security, few papers have addressed the relationship between risk-related variables and cost-effective network security decisions that impact the objective. Nevertheless, security measure selection problems have received some attention in more recent literature.

The problem of selecting the right security measures given a limited budget is clearly not an easy task. Most security planning models in the literature are qualitative and only few of them rely on quantitative approaches. In case of a pipeline network, the security risk assessment procedure elaborated by Reniers and Dullaert (2012) may be used. After a careful pipeline security risk assessment, the user is in possession of pipeline segment risk data as well as pipeline route risk data. Assuming that the security risk analyst determines a set of available security measures and defence strategies for application to the different pipeline segments and/or for the pipeline routes, a selection of the most effective security measures with respect to the available budget (either for a single pipeline segment or for a pipeline route) can be calculated. If the cost of the security measures is known in advance a mathematical approach can be used to solve the problem of optimal allocation of security resources by solving a knapsack problem. Reniers et al. (2012) explain how this well-known technique in the field of Operations Research is easy to use in case of security optimisation problems. A practical application to secure an illustrative pipeline infrastructure used to transport oil is described in Talarico et al. (in press 2014).

In Bistarelli et al. (2007) a method for the identification of the assets, the threats and the vulnerabilities of ICT systems is introduced. Furthermore, a qualitative approach for the selection of security measures to protect an IT infrastructure from external attacks is discussed. In particular, two security models based on defence trees (an extension of attack trees) and preferences over security measures are proposed.

In Viduto et al. (2012) the security of a telecommunication network is analysed from a quantitative point of view. Knowledge of potential risks enables organisations to take decisions on which security measures should be implemented before any potential threat can successfully exploit system vulnerabilities. A security measure selection problem is presented in which both cost and effectiveness of an implemented set of security measures are addressed. A Multi-Objective Tabu Search (MOTS) algorithm is developed to construct a set of non-dominated solutions, which can satisfy organisational security needs in a cost-effective manner.

In Sawik (2013) a similar security measure selection problem for an IT infrastructure is formulated as a single- or bi-objective mixed integer programming problem. Given a set of potential threats and a set of available security measures, the decision maker needs to determine which security measure to implement, under a limited budget, to minimize potential losses from successful cyber-attacks and mitigate the impact of disruptions caused by IT security incidents.

The prevention of heavy losses due to cyber-attacks and other information system failures in an IT network is usually associated with continuous investment in different security measures. In Bojanc and Jerman-Blažič (2008) several approaches enabling the assessment of the necessary investments in security technology are addressed from an economical point of view. The paper introduces methods for the identification of risks in ICT systems and proposes a procedure that enables the selection of the optimal level of investments in security measures.

Once security risks have been identified, the potential loss associated with their occurrence, as well as their probability of occurrence must be determined. Determining both probability of occurrence and potential impact of each risk is done in a process called *risk assessment*. Performing a risk assessment phase allows to take decisions regarding the necessary investment in security controls and systems. In our paper we assume that a preliminary risk assessment phase has been conducted by experts, in order to determine the probability of attacks associated with each network edge together with the costs and benefits of each available security measure.

Our approach extends the works of Reniers and Dullaert (2012) and Reniers et al. (2012) by defining a single-objective problem and proposing a quantitative method to select appropriate security measures. A different objective function is used, which relies on the minimization of the risk of the network to be not accessible between a couple of network nodes instead of the maximization of the effectiveness of the security measures used. Moreover, in our work, since a list of security measures is defined for each edge of the network, the model incorporates not only decisions taken at the level of the network, as done in Reniers and Dullaert (2012), Reniers et al. (2012) and Sawik (2013), but it depends on the choices made at the level of single network edges.

3 Problem description

The utility network can be represented by using a graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N}$ represents a set of nodes and $\mathcal{E}$ a set of edges, connecting the nodes. All edges $e_i \in \mathcal{E}$ have a probability of being attacked and failing, denoted as p_i. A set of security strategies $\mathcal{S}_i$, is defined for each edge $e_i \in \mathcal{E}$ and it comprises all security strategies s_{ij} that are available for this edge.

For each security strategy s_{ij} of edge e_i there are a cost c_{ij} and a value p_{ij}, which represents the probability of failure of this edge when s_{ij} is applied. Only one security

Table 1: Set of security strategies $\mathcal{S}_i$ for edge e_i

Strategy	Security measures	Cost	Probability
0	-	0	0.6
1	A	100	0.5
2	B	150	0.45
3	C	200	0.4
4	A&B	250	0.32
5	B&C	300	0.25

strategy per edge can be applied. A security strategy can be a combination of single 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 available due to their incompatibility.

The default security strategy s_{i0}, that has a cost $c_{i0} = 0$, is a default state that indicates that no security measures are applied. Its related probability p_{i0} represents the risk of edge e_i failure in case no security measure is selected. It represents the probability of failure of that edge given that an attacker is rational. The probabilities are unique for each edge, and are based on several information such as geographical location, length and criticality of that edge in the network. We assume that these probabilities are predefined in the risk assessment done by a security professional.

The model proposed in this paper makes the assumption that only edges can be attacked and that nodes are well protected and no viable target for an attack. In future research the model will be extended to a more general case where nodes are targets as well.

Given an origin node o and a destination node d in the network, the quality of a solution (i.e., a selection of a security strategy for each edge) is defined as the probability that no path exists between node o and node d. This would make it 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, if all connections to node d were unavailable).

In a communication network it is necessary that the whole network remains connected after an attack in order to guarantee a proper transfer of data between an origin node and a destination node. While extending the analysis to water/gas/electricity networks it is possible that after an attack a sub-network could still operate, but the transfer of a service/product between a supplier (our origin node) and a final user (our destination node) is not possible due to the lack of connections after an attack. In fact some edges that are not available due to an attack might disconnect the end user from the global utility network.

In this paper, since the decision problem is introduced for the first time, the problem is simplified by making the assumption that only one supplier and one customer exist in the network. However, several intermediate network nodes through which the service/goods

pass along the network to reach the customer are considered. In future work, this model will be extended as to evaluate several utility suppliers and several customers in the network, minimizing the risk that any customer is separated from any of the suppliers. In addition supplier capacity, customer demand and importance of either of them might be considered.

Given the fact that we have a single source node and a single destination node, in order to calculate the risk of the network being out of service, we make use of probability theory. Probability theory is used extensively in reliability theory and in reliability studies of systems. For an overview, we refer to Bazovsky (2004); Ministery of Defence (UK) (2011); Romeu (2004).

3.1 Mathematical model

In order to mathematically state the decision problem associated to the selection of the best set of security strategies to increase the overall network security, we first have to define the risk of network $\mathcal{G}$ being not available between source node o and destination d. For this reason we define a set $\mathcal{C}$. This set contains the combinations of edges that will disconnect all paths in the network between nodes o and d. In other words each element l of set $\mathcal{C}$ represents a combination of edges, contained in set $\mathcal{E}_l^F$, for which failure happens, and edges contained in set $\mathcal{E}_l^N$, which do not fail. It should be noted that $\mathcal{E}_l^F \cup \mathcal{E}_l^N = \mathcal{E},\ \forall l \in \mathcal{C}$. In addition, each element l in set $\mathcal{C}$ contains a critical combination of edge failures (from now on called scenarios). If the edges in $\mathcal{E}_l^F$ are out of service, a network breakdown between o and d is generated. The cardinality of set $\mathcal{C}$ depends on the topology of the network $\mathcal{G}$ and the position of nodes o and d within the network. Let B represents the available security budget and x_{ij} a binary variable, that takes values 1 when the security strategy j on edge i is applied, and 0 otherwise.

$$\min \quad \sum_{l \in \mathcal{C}} R_l \tag{1}$$

s.t.

$$\sum_{i \in \mathcal{E}} \sum_{j \in \mathcal{S}_i} c_{ij} \cdot x_{ij} \leq B \tag{2}$$

$$p_i = \sum_{j \in \mathcal{S}_i} p_{ij} \cdot x_{ij} \qquad \forall i \in \mathcal{E} \tag{3}$$

$$R_l = \prod_{i \in \mathcal{E}_l^F} p_i \cdot \prod_{k \in \mathcal{E}_l^N} (1 - p_k) \qquad \forall l \in \mathcal{C} \tag{4}$$

$$\sum_{j \in \mathcal{S}_i} x_{ij} = 1 \qquad \forall i \in \mathcal{E} \tag{5}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{E}, \forall j \in \mathcal{S}_i \tag{6}$$

The objective function (1) minimizes the total risk of the network being out of service between nodes o and d. The total network risk is given by the sum of risks associated to single scenarios happening. Constraint 2 ensures that the total cost associated to the selected security strategies does not exceed the predefined security budget B. Equation 3 is used to define the probability p_i associated to a failure of edge e_i. Equation 4 gives us the risk of a scenario happening, which disconnects the paths in the network between nodes o and d. Equation 5 forces the decision process to select at maximum one security strategy to protect edge e_i. It should be noted that $x_{i0} = 1$ means that for edge e_i no security measures have been applied. Finally, constraint 6 represents the domain of the decision variable, which ensures that no partial security strategies are allowed.

3.2 Illustrative Example

To clarify the computation used to determine the total risk of the network being down between nodes o and d, we report the following example shown in Figure 1.

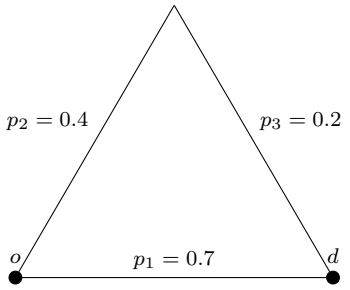

Figure 1: Utility network $\mathcal{G}$ with a source o and a destination d. On each edge the associated probability of an event happening is reported

In order to simplify the notation and the computation of the risk of network $\mathcal{G}$ being down, we suppose that no security strategy is available for the edges in the network that we named edges 1, 2 and 3. Given the topology of the network, three different combinations of events might disconnect the network between nodes o and d. These cases have been reported in Table 2. For example, the first combination implies that, if both edges 1 and 2 are not available because an event has happened, the service can't be delivered to node d from node o, also if no event will happen on edge 3. The probability associated to this combination of events is equal to $p_1 \cdot p_2 \cdot (1 - p_3)$ that is $0.4 \cdot 0.7 \cdot (1 - 0.2) = 0.224$. The same reasoning applies to the remaining critical combinations reported in Table 2. Starting from the value of R_l it is possible to compute the total risk of the network being down, which in this case is equal to 0.364.

l	$\mathcal{E}_l^F$			$\mathcal{E}_l^N$	R_l
1	1	2	-	3	0.224
2	1	3	-	2	0.084
3	1	2	3	-	0.056

Table 2: List of critical events contained in $\mathcal{C}$ for network $\mathcal{G}$

4 Solution approach

The decision problem of selecting appropriate security strategies given a budget constraint, in order to reduce the risk of the utility network being down, belongs to the more general category of knapsack problems.

The single objective knapsack problem is one of the best known combinatorial optimisation problems. This problem can be described as follows: given a set of n items, each with a certain weight w_i and a certain profit p_i with $i \in [1, n]$, the objective is to select the subset of items of which the total profit is maximal, and the total weight does not exceed the knapsack capacity C.

Applications of the knapsack problem are frequently encountered in several real-world decision-making processes in different fields such as portfolio management, menu planning and design of experiments. For a detailed review of the knapsack problem, the reader is referred to Wilbaut et al. (2008).

Our problem, since the objective function is not linear, belongs to the class of non-linear knapsack problems, also known as the non-linear resource allocation problem. This problem also belongs to the category of combinatorial optimisation problems (see Bretthauer and Shetty (2002)). As the problem instances grow larger, an exact algorithm will require an exponential amount of time. Therefore, we decided to sacrifice the optimality for near optimal solutions that can be calculated in a very short amount of time. To achieve this goal we will make use of metaheuristics.

The metaheuristic that has been developed in this paper is shown in Algorithm 1. It belongs to the category of iterated local search algorithms (ILS). The reader is referred to Lourenço et al. (2010) for a recent survey on ILS.

More specifically our ILS metaheuristic is composed of three consecutive steps: (1) a greedy random adaptive search procedure (GRASP) is used during the constructive phase (see Section 4.1) to generate an initial solution; (2) a variable neighbourhood descent (VND) is used during the improvement phase (see Section 4.2) to improve the current solution and; (3) two perturbation heuristics are used during the diversification stage (see Section 4.3) 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.

Algorithm 1 Metaheuristic structure

```
Initialize both Problem and Heuristic parameters
let x be the current solution and f(x) its cost
let x* be the best solution found so far and f(x*) its cost
x ← GRASP Heuristic()
x* ← ∅, f(x*) ← ∞
while (max number of iterations not reached) do
   x ← Improvement(x)
   if (f(x) < f(x*)) then
      x* ← x, f(x*) ← f(x)
   end if
   update number of iterations without improvement
   if (max number of iterations without improvement not reached) then
      x ← Perturbation(x)
   else
      x ← GRASP heuristic()
   end if
   update number of iterations
end while
return x*
```

The first step of this iterative solution approach consists of running a GRASP constructive heuristic that selects promising edges, and selects from that set of promising edges the best security strategy. This selection is repeated until the security budget does not allow any further security strategies.

After the GRASP procedure is finished, we use local search to improve the current solution by using a VND heuristic. This local search is executed until the algorithm finds no more improvement. Once this is the case, a perturbation is applied to escape the local optimum, and the algorithm continues with a local search from this perturbed solution. If after a predefined number of perturbations no better solution can be found, the algorithm is restarted from a new solution generated by the GRASP constructive heuristic.

4.1 Constructive phase

Our solution approach uses a greedy randomized adaptive search procedure (GRASP) to generate an initial solution x.

GRASP is a well-known constructive heuristic (see Feo et al. (1991) for more details), that, starting from an empty solution, generates a complete solution by adding one strategy at a time until either the security budget is consumed or no security strategy is available.

At each iteration of the GRASP heuristic the strategy to be added in the current solution is randomly selected from a restricted candidate list (RCL). The size of the RCL, `alpha`,

is a parameter that controls the balance between greediness and randomness. If `alpha` is large, the selection is relatively random, while if `alpha`= 1 the greedy construction is completely deterministic.

In our approach, *RCL* contains the first `alpha` *critical edges* ordered by decreasing value of criticality for which: (a) no countermeasure has previously been included in the current solution and (b) at least one countermeasure, not contained in the tabu list and whose cost is lower than the remaining budget, is available.

The criticality of an edge is a value measuring the importance of that edge for the security of the network. The higher the criticality, the greater the impact of that edge on the risk reduction and thus on the reduction of the objective function. In practice the number of times that that edge occurs in the critical combinations multiplied by the remaining probability of that edge, after the application of a security strategy, is used as an estimator of criticality.

In this way, at each iteration of the GRASP heuristic the list *RCL* contain the first `alpha` edges that can likely generate higher risk reductions considering the overall network. At each iteration of the GRASP heuristic an edge e is randomly chosen from *RCL* and an available strategy s whose cost is lower than the remaining budget is selected for that edge. During the constructive method a tabu list is used in order to exclude the selection of strategies that have already been explored in previous iterations of the metaheuristic. Two possible selection mechanisms can be used so select a strategy s for edge e at each iteration of the heuristic:

- *Best improvement* which is more demanding in terms of computation time , but provides at each iteration the usage of the combination (e, s) which yields the highest reduction in the objective function given all the considered options;
- *First improvement* which guarantees a fast random selection of a feasible (from the remaining budget point of view) combination (e, s), which is not used in the current solution and is not contained in the tabu list. This combination can provide a reduction of the risk for the overall network.

After the selection process the current solution, the remaining budget and the list *RCL* are all updated. Selected edge e is removed from *RCL* as well as all the edges for which no strategy is available any longer due to a lower remaining budget. Then the list *RCL* is reordered based on the new vulnerability values associated to the first `alpha` critical edges. The GRASP heuristic is repeated until either the remaining budget is still positive or the list of available options is empty. An overview of the GRASP constructive heuristic is reported in Algorithm 2.

4.2 Improvement phase

The improvement of the current solution is performed by a variable neighbourhood descent (VND) heuristic. The VND is a deterministic variant of the well-known variable

Algorithm 2 GRASP constructive heuristic

```
Initialize tabu list TL and remaining budget R_B = B
compute the vulnerability for each edge
let x ← {∅} be the current solution
while ( R_B > 0 && RCL! = ∅) do
   select an edge e from RCL
   select a strategy s for edge e
   add (e, s) to x
   update R_B
   update RCL
   update TL
end while
return x
```

neighbourhood search (VNS) metaheuristic (Hansen and Mladenović, 2001). In general VNS algorithms use a sequence of nested neighbourhood, $\mathcal{N}_1, \ldots, \mathcal{N}_{k_{max}}$ with an increasing size, i.e., $\mathcal{N}_k \subset \mathcal{N}_{k+1}$ and a perturbation move is used for diversification purposes.

In our algorithm diversification is introduced by applying the VND heuristic to a different current solution. This solution is generated either after the above mentioned GRASP constructive heuristic (at the first iteration of the ILS metaheuristic) or after the perturbation phase (see Section 4.3. The structure of the VND heuristic developed in this paper is shown in Algorithm 3.

Algorithm 3 VND Heuristic

```
Let x be the current solution and f(x) its cost
k ← 1
while (k < k_max) do
   x' ← N_k(x)
   if (f(x') < f(x)) then
      x ← x'
      k ← 1
   else
      k++
   end if
end while
return x
```

The algorithm uses three neighbourhood structures. The first one, called *Internal Swap*, attempts to replace a strategy s for a given edge e with another strategy s', that is not contained in the tabu list, for which the remaining budget plus the difference in cost between s and s' is non negative.

The second neighbourhood structure, called *External Swap*, attempts to replace one strategy s for edge e with another strategy s', that is not in the tabu list, associated to a different edge e' not previously considered in the current solution. In practice part

of the security budget is transferred from edge e to edge e' with the cost of strategy s' lower than the remaining budget plus the cost of strategy s'.

The third neighbourhood structure, called *Double Swap*, is a variant of the second move where two *External Swap* moves are executed at the same time. In practice strategies s and s' associated to edges e and e' respectively are simultaneously swapped with other two strategies $\bar{s}$ and $\bar{s}'$ available for edges e and e'. It should be noted that the new strategies $\bar{s}$ and $\bar{s}'$ must not be contained in the tabu list and the cost of the strategies $\bar{s}$ and $\bar{s}'$ must not be greater than the remaining budget plus the cost of the removed strategies s and s'.

Moreover, to speed up the improvement stage only a restricted number of edges are considered and only moves which have a positive contribution to the current solution are executed. In particular, the *External Swap* attempts to replace a strategy s associated to any edge e already in the solution with one of the available strategies associated to the first `alpha` edges, not used in the solution, that are ordered by decreasing value of criticality, as described before.

The size of the neighbourhood that *Double Swap* has to explore is equal to $\bar{\mathcal{A}} \times (\bar{\mathcal{A}} - 1)$ where $|\bar{\mathcal{A}}|$ is the amount of edges used in the current solution. This number depending on the solution can be quite extensive to be explored. For this reason the number of evaluations that are performed by the *Double Swap* is restricted to a maximum value based on a percentage of set $\bar{\mathcal{A}}$.

In addition, also in the VND heuristic, two different selection mechanisms can be used to improve the current solution. The first one is based on a *first improvement* mechanism while the second one relies on a *best improvement* mechanism. The VND heuristic ends when no improvements of the current solution can be found.

4.3 Diversification phase

The diversification phase attempts to escape from local optima, after the improvement stage, by exploring different areas of the search space hopefully not yet explored. Two different diversification mechanism are used in our ILS metaheuristic: (1) If a maximum number of iterations without having obtained any improvement of the current solution is not reached a perturbation heuristic is used; (2) If the algorithm does not manage to find any further improvement, a new initial solution is generated by using the GRASP constructive heuristic.

The perturbation heuristic partially "destroys" the current solution by removing a number of strategies that have been used. These strategies are inserted in a tabu list in order to avoid the reuse of the same strategies in other solutions for a given number of iterations. The perturbation heuristic, by removing strategies, allows to free some budget resources, making room for new, unused strategies that are not in the tabu list. These new strategies are selected using either a "best improvement" or a "first improvement"

selection mechanisms and added to the current solution as long as budget is available and at least one strategy, with a cost less than the remaining budget and not contained in the tabu list, is available. The newly generated solution becomes the input of the VND heuristic for further improvement.

When the number of iterations without having obtained any improvement of the current solution is reached, in order to explore a new area of the search space, a new solution is generated from scratch by using the GRASP constructive heuristic, described before, which selects feasible strategies that are not contained in the tabu list.

5 Computational experiment

In this section, the stages used to execute our computational experiments are described. First, some realistic instances are generated and used later on for tuning and testing the metaheuristic described before (see Section 5.1).

In the second stage, the parameters for the solution approach are tuned in order to achieve the best results (see Section 5.2).

Finally, the metaheuristic with its "best" parameter setting is used to solve the test instances and computational results are shown in Section 5.3.

5.1 Instance generation

To test the algorithm developed in this paper a library of test instances was created. All instances used in this paper can be found on `http://antor.uantwerpen.be/downloads`. The software that was developed to generate the instances, takes several parameters as input, necessary to specify the properties of the instances. These parameters are: (1) The maximum budget that can be spent on strategies; (2) The number of nodes in the network; (3) The maximum number of strategies for each edge that connects two nodes; (4) The percentage of edges of the Delaunay triangulation that should be added to the minimum spanning tree, to add some redundancy as explained below; (5) The number of instances to be generated for each set of parameters.

A first step in the instance generation process is the generation of nodes. The node's coordinates are randomly selected from a uniform distribution between 0 and 100. This range can be adjusted by the decision maker. After the nodes are generated, a Delaunay triangulation (Delaunay, 1934) of this set is created. The Delaunay triangulation is used to identify the "neighbours" of each node, which will be used to create the final network. When the neighbours of all nodes are identified, Kruskal's algorithm (Kruskal, 1956) is used to generate a minimum spanning tree. The motivation for starting from a minimum spanning tree is that this is the most cost-effective way to connect all nodes in the network.

In real life cases, service providers often add redundancy to increase operational security. If one edge gets disabled, customers are serviced trough the redundant edge. For this reason, the minimum spanning tree is extended with edges to mimic the redundancy that is present in real life networks. This is done by selecting a percentage of unused shortest edges, based on the parameter settings.

When the edges of the network have been generated, each edge is assigned a random probability of attack.

When each edge has been assigned a probability, the algorithm generates a random number of strategies with a minimum of one, and a maximum number as defined in the parameters. Each strategy is assigned a random reduction in probability of attack between 1 and 20 percent, and a random cost based on this percentage reduction. To obtain this cost, this reduction is multiplied with a random number between 0.5 and 1.5 and the maximum budget, to get realistic values.

5.2 Parameter tuning

The metaheusritic described before uses some internal parameters that need to be tuned in order to have a good compromise between performance and quality of the solutions. The internal parameters have been described before in Section 4 and are summarized in Table 3 together with the tested values.

Table 3: Heuristic parameters

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

The parameter tuning has been conducted on a set of instances consisting of smaller networks of 10 nodes (10 - 12 edges) and larger networks of 18 nodes and 20 edges.

A full factorial statistical experiment was conducted trying the values reported in Table 3. The objective values and times are averaged, and shown in Figure 2. The parameters with an asterisk had a significant effect on both the objective value and time, while parameter `tabu-tenure` was significant only in regard to the time. Parameter

`max-iter-no-improvement` was not significant, but looking at the time and the objective value profiles, a parameter setting equal to 10 has been selected.

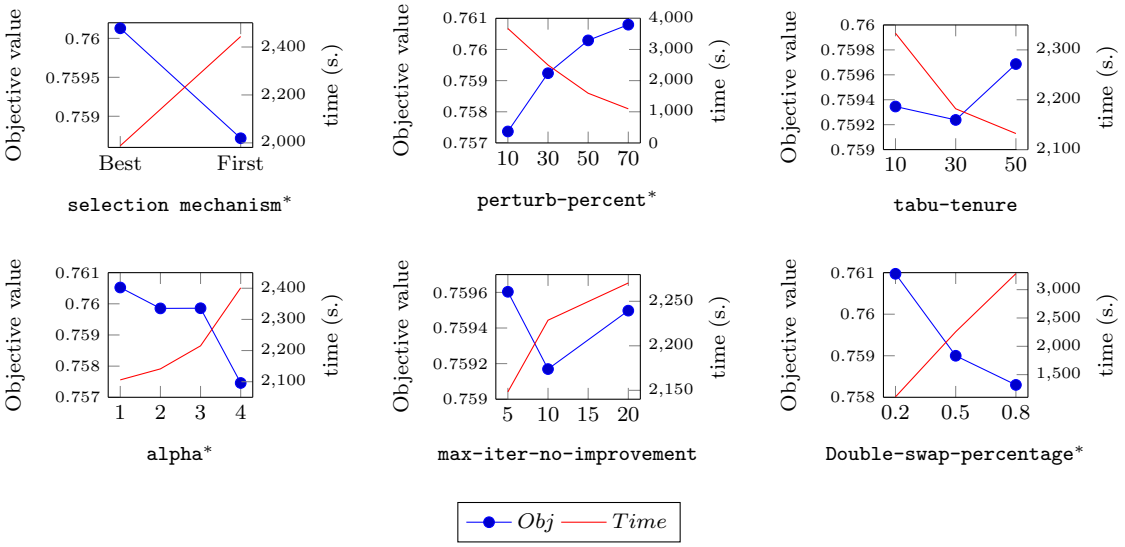

Figure 2: Plot of average objective values and times for given parameter setting

Looking at the minimisation of the objective function the chosen settings for the metaheuristic parameters are shown in Table 4.

Table 4: Selected heuristic parameters

Parameter	**Setting**
`max-iter`	50
`perturb-percent`	10%
`alpha`	4
`tabu-tenure`	30
`max-iter-no-improvement`	10
`Double-swap-percentage`	80%
`selection mechanism`	first

5.3 Computational results

After having selected the "best" parameters settings for the solution approach, we analyse the influence of each metaheuristic component on both the quality of the solutions and the running time on instances ranging from 9 up to 20 edges and 5 up to 20 security strategies per edge.

In order to analyse the quality of the solutions, the results found by the proposed metaheuristic are compared with the optimal solutions obtained by an exact approach. The exact approach was based on an exhaustive exploration of the search space were the

budget constraint was used as a cutting plane to speed up the running time. In Table 5 the comparison between the metaheuristic and the exact approach is reported. The metaheuristic has been run 25 times on each instance to compute the average running time and objective value that are reported in Table 5 together with the best solution found over these runs. The metaheuristic obtained 9 optimal solutions (highlighted in bold) out of 16 test instances. The percentage optimal gap, used as an indicator of the quality of the obtained solutions, is around 0.51%. Considering all the solutions averaged over 25 runs the gaps from the optimal solutions is only 1.62%.

Looking at the running times needed by the exact approach, it explodes when increasing the number of edges in the network from 8 to 11, while the running time required by the metaheuristic remains stable. Analysing both the exact approach and the metaheuristic, it becomes clear that the proposed metaheuristc achieves good results in a short amount of time. On average the solution approach, proposed in this paper, was 3880 times faster than the exact approach.

	Exact Approach		**Metaheuristic**				
Instance	Optimal Solution	Time(s.)	Best Solution	Average Solution	Average Time (s.)	Best Gap	Average Gap
NS-n8-c5-C3-a30-x0	0.64813	636.910	0.65185	0.65468	0.147	0.57%	1.01%
NS-n8-c5-C3-a30-x1	0.81428	37.275	0.81428	0.84409	0.614	**0.00%**	3.66%
NS-n8-c5-C3-a30-x2	0.18171	104.381	0.18483	0.18549	0.642	1.71%	2.08%
NS-n8-c5-C3-a30-x3	0.13410	57.631	0.13711	0.14310	0.573	2.24%	6.71%
NS-n8-c5-C3-a30-x4	0.64803	54.456	0.65369	0.66393	0.439	0.87%	2.45%
NS-n8-c5-C3-a30-x5	0.34060	13.512	0.34060	0.34333	0.504	**0.00%**	0.80%
NS-n8-c5-C3-a30-x6	0.25193	3.372	0.25833	0.26179	0.697	2.54%	3.91%
NS-n8-c5-C3-a30-x7	0.07905	65.642	0.07905	0.07947	0.549	**0.00%**	0.54%
NS-n9-c5-C3-a30-x0	0.89334	1356.729	0.89442	0.89842	0.527	0.57%	0.12%
NS-n9-c5-C3-a30-x1	0.82052	5744.238	0.82052	0.82149	0.702	**0.00%**	0.12%
NS-n9-c5-C3-a30-x10	0.41173	222.962	0.41173	0.42319	0.218	**0.00%**	2.78%
NS-n9-c5-C3-a30-x11	0.07290	588.678	0.07290	0.07290	0.254	**0.00%**	0.00%
NS-n9-c5-C3-a30-x12	0.79698	1160.176	0.79700	0.79754	0.414	**0.00%**	0.07%
NS-n9-c5-C3-a30-x14	0.11105	7825.799	0.11105	0.11123	0.503	**0.00%**	0.16%
NS-n9-c5-C3-a30-x4	0.40821	3178.317	0.40821	0.40962	0.304	**0.00%**	0.34%
NS-n9-c5-C3-a30-x8	0.43050	8751.806	0.43067	0.43340	0.594	0.04%	0.67%
Average		**1862.618**			**0.480**	**0.51%**	**1.62%**

Table 5: Exact approach in comparison with the metaheuristic for instances with 8 nodes and 9 edges (instances name with the code n8) and with 9 nodes and 11 edges (instances name with the code n9) and 5 security strategies per edge

Bigger instances have also been solved by using an exact approach, but the computation time drastically increased. For an instance having 13 edges, the exact approach took around 3 days to find the optimal solution. The solution found by the heuristic was only 1.77% far from this optimal value. This near optimal solution has been achieved in 0.475 seconds, that resulted significantly smaller than the time needed by the exact approach. When the amount of edges and security strategies rise even more, it will not be practical to solve larger instance with an exact approach. For this reason, we decided to use a time limit of 5 hours on the exact solution method, to find a good upper-bound. This value is used in Table 6 to evaluate the quality of the results provided by the metaheuristic.

It is worth noticing that the solution obtained by using a time limit for the instance NS-n10-c5-C3-a30-x2 (shown in Table 6) resulted in a good upper bound, presenting only

a difference of 0.11% from the optimal known solution. This means that the selected time limit is adequate, for that instance, to achieve a good upper-bound to be used to analyse the quality of the solutions obtained by the metaheuristic.

Some results are shown in Table 6, where on average the metaheuristic improved the level of security of the initial network by 32%. In eight cases, the metaheuristic outperformed the exact approach with the time limit, finding better results in a short computation time.

		Exact Approach		**Metaheuristic**				
Instance	Original risk	Upper-bound	Time(s.)	Best Solution	Average Solution	Time(s.)	Best Gap	Average Gap
NS-n10-c5-C3-a30-x0	0.99998	0.91833	18000	0.91276	0.91320	0.614	-0.61%	-0.56%
NS-n10-c5-C3-a30-x1	0.99996	0.20114	18000	0.16652	0.16743	0.374	-17.21%	-16.76%
NS-n10-c5-C3-a30-x2*	≈1	0.86129	18000	0.87553	0.88744	0.475	1.65%	3.04%
NS-n10-c5-C3-a30-x3	≈1	0.58132	18000	0.55206	0.55669	0.453	-5.03%	-4.24%
NS-n10-c5-C3-a30-x4	≈1	0.64521	18000	0.64708	0.67119	0.353	0.29%	4.03%
NS-n18-c10-C3-a30-x0	≈1	0.99675	18000	0.98455	0.98461	438.199	-1.22%	-1.22%
NS-n18-c10-C3-a30-x1	≈1	0.61578	18000	0.55764	0.55771	207.571	-9.44%	-9.43%
NS-n18-c10-C3-a30-x2	≈1	0.94620	18000	0.90084	0.90093	363.550	-4.79%	-4.78%
NS-n18-c10-C3-a30-x3	≈1	0.40736	18000	0.31192	0.31198	343.673	-23.43%	-23.41%
NS-n18-c10-C3-a30-x4	≈1	0.93503	18000	0.84476	0.84497	255.071	-9.65%	-9.63%
Average	≈1			**0.67536**	**0.67962**			

Table 6: Results for instances in the range 10-20 edges and 5-20 security strategies per edge

In Figure 3 we report the evolution of the objective function associated with both the best and the current solutions over time. It can be noticed that our solution approach is able to converge towards good results in a short CPU time also in case of large instances. It should be noted that the marginal improvement of the best solution found so far significantly decreases when the running time is increased.

Analysing the plot associated with the current solution in Figure 3, should clarify the behaviour of the solution approach on the quality of the solution. After the perturbation, that destroys the quality of the solution, small improvements obtained during the VND heuristic can lead to better solutions. One can clearly distinguish the perturbation strategy that allows the algorithm to efficiently escape from local optima. Starting from the perturbed solution, presented by a higher objective value (thus denoted with a pick), the VND heuristic guides the current solution trough small improvements towards a new local optimum and a hopefully a new and better solution becomes available. The fact that the VND heuristic is able to decrease the value of the perturbed solution and detects new local optima prove the efficiency of the VND.

We also analysed the effect of each metaheuristic component on the objective value. The VND heuristic on average can improve the initial solutions generated by the GRASP constructive heuristic by 1%. Moreover the perturbation is quite effective since it can efficiently aid the solution approach to escape from local optima and improve the quality of solutions on average by 0.14%.

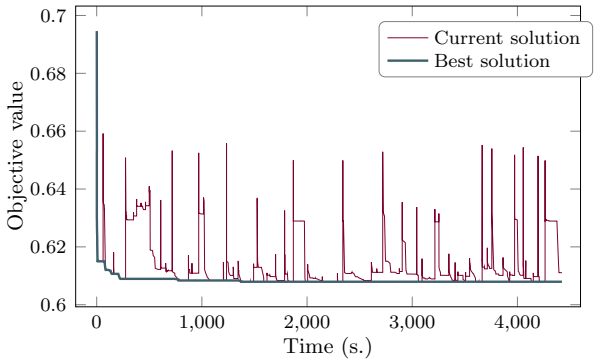

Figure 3: Plot of the objective value over time.

6 Conclusion

In this paper we describe a model for the selection of the appropriate security strategies given a limited budget to increase the security of an infrastructure such as pipelines transportation systems, telecommunication networks and smart grids.

By selecting an origin node and a destination node, it is possible to define the risk associated to the interruption of service (or material flow) due to external malicious attacks (e.g., terrorism, vandalism) directed at the destruction of one or several edges.

Redundancies in the network might be used by the service provider or network owner in order to increase the network reliability, keeping the service available in case of problems affecting a single edge of the network. However, attacks directed at critical edges might disrupt the complete infrastructure and service.

An exact evaluation of the risk of the whole network being unavailable might be a difficult task, especially when several loops are present inside the network. In order to reduce the complexity of computations, we defined an approach to have an accurate estimate of the risk of the network being down.

This method considers the global impact of a limited number of edges that are unavailable at the same time and therefore might disconnect the origin node from the destination node. In order to prevent such episodes that could induce significant economic losses, security strategies can be implemented to increase the reliability of each network edge.

We assume that each edge presents different characteristics in terms of vulnerability to external attacks due to internal and external factors such as geographical location, length, materials used and operating conditions. In addition, we assume to have for each edge a list of security strategies, each one with different characteristics in terms of cost and effectiveness. The decision support model, presented in this paper, attempts to define an ideal mix of such strategies in order to increase the security of the overall

network while respecting the budget. This budget might restrict the choice between security strategies.

The decision model considered in this paper addresses multilevel decisions, since a decision made at the level of a single edge might affect the security of the whole network. We proposed an iterated local search heuristic, which exploits the benefits offered by tabu search hybridised with a GRASP and VND heuristic, to solve this combinatorial optimisation problem.

The ILS heuristic has been tuned in order to find its "best" configuration when minimising the objective value. In a second stage of the computational experiments, the tuned metaheuristic has been employed to solve a set of test instances that mimic possible realistic scenarios and the performance of the algorithm is evaluated.

Future research could be aimed at extending the model to support more realistic scenarios where e.g. the nodes are vulnerable and the connection of all nodes in the network should be ensured.

Acknowledgements

This research was partially 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. ISBN 9780486438672.

S. Bistarelli, F. Fioravanti, and P. Peretti. Using CP-nets as a guide for countermeasure selection. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 300 – 304, March 11 - 15, Seoul, Republic of Korea, 2007. ACM.

R. Bojanc and B. Jerman-Blažič. An economic modelling approach to information security risk management. *International Journal of Information Management*, 28(5):413 – 422, 2008.

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

B. N. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, 6:793 – 800, 1934.

T. Feo, J. Bard, and K. Venkatraman. A *GRASP* for a difficult single machine scheduling problem. *Computer and Operations Research*, 18(8):635 – 643, 1991.

P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449 – 467, 2001.

J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48 – 50, 1956.

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.

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, W. Dullaert, A. Audenaert, B. Ale, and K. Soudan. Managing domino effect-related security of industrial areas. *Journal of Loss Prevention in the Process Industries*, 21(3):336 – 343, 2008.

G. L. 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.

START: A Center of Excellence of the U.S. Department of Homeland Security. The Global Terrorism Database (GTD). 2014. `http://images.fedex.com/us/about/today/access/GreaterAccessChange_global.pdf`, visited on 2014-10-29.

L. Talarico, K. Sörensen, G. Reniers, and J. Springael. Pipeline security. In S. Hakim and D. C. Y. Shiftan, editors, *Securing Transportation Systems*. Springer Science and Business Media, LLC, New York, in press 2014.

V. Viduto, C. Maple, and D. Huang, W.and López-Peréz. A novel risk assessment and optimisation model for a multi-objective network security countermeasure selection problem. *Decision Support Systems*, 53(3):599 – 610, 2012.

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.