An Ant Colony Optimization Approach for Safest Path Pair Computation under Correlated Failures

Zoltán Tasnádi[1] [0000−0001−8127−9493], Balázs Vass[1,2] [0000−0002−8589−7165], and Noémi Gaskó[1] [0000−0002−3915−8986]

[1] Babeş-Bolyai University, Cluj Napoca, Romania
[2] Budapest University of Technology and Economics, Budapest, Hungary
{zoltan.tasnadi, balazs.vass, noemi.gasko}@ubbcluj.ro

Abstract. Finding the safest pair of paths between two specified endpoints s and t while accounting for multiple correlated failures is a complex computational challenge with various practical applications. In communication backbone networks, for instance, establishing a secure pair of paths between s and t is essential for meeting the high availability standards required by emerging technologies such as autonomous driving, AR/VR applications, or telesurgery. This paper first provides a formal proof of the $\mathcal{NP}$-hardness of the task. Then, we introduce the Safest Path Pair Ant Colony Optimization (SPP-ACO) algorithm. This new algorithm is based on the Max-Min Ant System. Numerical tests carried out on real-world datasets demonstrate the proposed method's effectiveness. The proposed SPP-ACO algorithm typically provides at least as safe paths as the baseline, even outperforming it in a significant share of the parameter settings. This grants a place for the SPP-ACO on the stage of best solutions for safest path pair computation in the presence of correlated failures.

Keywords: safest path pair · Ant Colony Optimization · correlated failures · Shared Risk Link Groups.

1 Introduction

The study of computational network problems has garnered significant interest in recent decades due to their vast range of applications, such as critical node detection in wireless ad hoc networks [21] or influence maximization in online advertisements [13]. A concrete, well-explored issue is the challenge of finding the shortest path between two nodes, s and t, within a graph G. When the objective is to compute an st-path with the fewest edges, a straightforward Breadth-First Search (BFS) can be employed. For graphs with non-negative edge weights, Dijkstra's algorithm [7] is the most effective solution, while the Bellman-Ford algorithm [2] is well-suited for graphs with arbitrary weights. Going a step further, one could search for a cheap pair of st-paths. For example, supposing nonnegative lengths, finding a link or node-disjoint st-path pair of minimum total length, one can use the highly scalable Suurballe's algorithm [25], which is practically just as fast as a simple Dijkstra's. Turning from shortest (or cheapest) paths to *safest st*-paths, one can check that, in certain simple scenarios, they can be computed by utilizing a shortest path finding subroutine, with some necessary transformations. E.g., paper [32] explores such fortunate settings, one of these being when the network element failures are independent of each other. However, the situation becomes less clear when these failures are correlated, the general case being $\mathcal{NP}$-hard [28]. Based on this, it is hardly surprising that finding a safest *pair* of *st*-paths in the presence of correlated network element failures is also a computationally hard problem (see Thm. 1). The primary objective of this study is to develop an efficient algorithm for finding a safest path pair in the presence of correlated failures. Although our primary numerical inputs are derived from communication backbone networks paired with seismic hazard data, we believe that the algorithm proposed in this paper can be effectively applied to a broad spectrum of problem inputs.

Safest paths and path pairs in backbone networks: Evaluating availability between two network nodes assuming independent single-element failures has a long history [34,17,23,1]. Dealing with multiple failures in communication networks has its traditions, relying on the concept of Shared Risk Groups (SRGs) (see [31] and references therein). Probabilistic extensions of SRLGs were also investigated [12,14,27]. We will use the unified terminology on probabilistic SRLGs proposed by [30]. Much of the work in this field tackled disaster modeling more heuristically in their own way to address their given problem in network planning. A more principled natural approach (also taken by this paper) is to take the disaster scenarios as input [16], that have been carefully precomputed by dedicated approaches, e.g., based on historical hazard data. By now, efficient methods for computing and storing the correlated link failures are available [30]. These methods are already in use in complex frameworks for disaster resilience [18,19]. From an algorithmic point of view, several variants of the problem of ensuring high availability, and different solution concepts were proposed [33,10,5]. Notably, [3] gives a highly scalable algorithm, that, under specific circumstances, finds a maximal number of SRG-disjoint paths. On the other hand, [4] proves the $\mathcal{NP}$-hardness of a similar problem formulation. Note that these latter problem formulations fail to handle algorithmically the case when there is no two SRG-disjoint *st*-path (thus, the optimal availability of such a path pair necessarily being lower than 1). Zooming out again, though considering link correlations poses extra computational challenges, it can significantly better the service availabilities and performance.

Nature-inspired algorithms [9] are novel problem-solving tools designed for hard optimization problems.

Regarding the safest path pair problem studied in this article, to the best of our knowledge, nature-inspired algorithms have not been investigated so far.

Ant Colony Optimization algorithm (ACO), designed for combinatorial optimization problems, is a good choice to solve network-related problems. For the safest path problem, an ACO was adapted by the same authors [28].

State-of-the-art works regarding ACO study different variants of the shortest path problems: [15] describes an ant colony system for the shortest path problem, where preferred edges can be used. In [11] an ACO is designed for the stochastic shortest path problem, where edge weights can be noisy. The work [6] studies the shortest path problem with fuzzy weights, presenting a fuzzy-based ACO.

The main contributions of the paper are as follows:

- We formalize and prove the $\mathcal{NP}$-hardness of the problem of finding a safest path pair in presence of correlated failures.
- We adopt the Ant Colony Optimization algorithm (ACO) to solve the problem.

The rest of the paper is organized as follows: in Sec. 2 the problem is defined and $\mathcal{NP}$-hardness is proven. Sec. 3 presents the proposed solving method, the Safest Path Pair Ant Colony Optimization algorithm. The next Section presents the numerical experiments conducted on real-world networks. Finally, Sec. 5 concludes the paper, and some future research possibilities are presented in this Section.

2 Problem statement and computational hardness

2.1 Problem definition

The problem input consists of two main parts. One is a connected graph $G = (V, E)$, along with a communication source-target node pair $\{s, t\} \subseteq V$. The other part of the problem input encodes the probabilities of joint failures of link sets. For this, for a link set $S \subseteq E$, in line with [30,19], we define $\text{FP}(S)$ (that stands for 'link failure state probability of S') to denote the probability that *exactly* link set S will fail at the next disaster. The second part of the input is $\text{FP}[G]$, which is a data structure containing all the $\text{FP}(S)$ values, where we list $\text{FP}(S)$ only if $\text{FP}(S) > 0$. Note that in most of the natural settings, $\text{FP}[G]$ has a manageable size [30]. We note that albeit $\text{FP}[G]$ stores only link failures, it is suitable for implicitly storing node failure probabilities, too; see [30, Sec. V.]. The goal is to find a safest path pair among a node pair s and t, i.e., a pair of st-paths P_1 and P_2 such that the chance of both P_1 and P_2 being cut by the next disaster is minimal.

Below, we give a more formal definition of the above concept, followed by a proof of $\mathcal{NP}$-hardness of the decision version of the safest path problem. FP can be defined as follows.

Definition 1 (Link Failure State Probability (FP)). *Given a link set $S \subseteq E$, the link failure state probability (FP) of S, denoted by FP(S), is the probability that* exactly *the links of S fail simultaneously (and no other links).*

Note that if there is a disaster scenario that leads a set of links S to be listed in $\text{FP}[G]$, then all the subsets of S are subject to possible common failures. Since there are some optimization approaches that, for each link set T, take as input the probabilities that *at least* the links in T will fail (and possibly some others too), we will define the *cumulative failure probability* of link sets:

Definition 2 (Cumulative Failure Probability (CFP)). *Given a link set $S \subseteq E$, the cumulative failure probability (CFP) of S, denoted by CFP(S), is the probability that all links in S fail simultaneously (and possibly other links too).*

Again, it is easy to see that if for a link set S, $\mathrm{CFP}(S) > 0$, then all the $2^{|S|} - 1$ non-empty subsets of S have to be stored in $\mathrm{CFP}[G]$. Whenever it does not cause confusion, we will refer as 'FP' to both 1) the tuple $(S, \mathrm{FP}(S))$ for a link set S, and 2) simply, to $\mathrm{FP}(S)$. The same goes for 'FP'. Intuitively, $\mathrm{FP}[G]$ and $\mathrm{CFP}[G]$ are interconnected in a similar way to the density and cumulative density functions. Next, we define two interconnected versions of the safest path pair finding problem with correlated link failures:

Problem 1: Safest Path Pair FP Problem - decision version

Input: A graph $G = (V, E)$, nodes s and t, a threshold T, and failure probabilities $\mathrm{FP}[G]$.

Question: Decide whether there exists such an st-path pair P and Q, from which, after the next disaster, at least one remains intact with a probability of at least T.

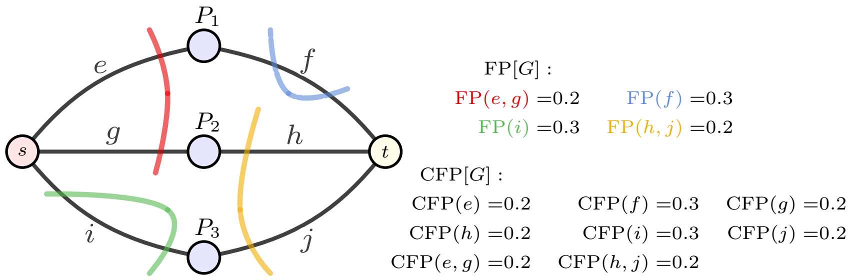
Problem 2: Safest Path Pair CFP Problem - decision version

Input: A graph $G = (V, E)$, nodes s and t, a threshold T, and failure probabilities $\mathrm{CFP}[G]$.

Question: Decide whether there exists such an st-path pair P and Q, from which, after the next disaster, at least one remains intact with a probability of at least T.

As we will see in Thm. 1 both problem variants are $\mathcal{NP}$-hard. Consequently, optimizing the availability of an st-path pair is a computationally hard problem. But first, we depict the following example problem input.

Example 1. Fig. 1 depicts a simple example of the problem inputs, along with the availabilities of each of the st-paths and st-path pairs. This example showcases that a safest st-path may not be eligible in an st-path pair of highest availability. In our case, despite that the availability of $P_2 = \{g, h\}$ being the maximal, standing at 0.6 (exceeding the availabilities of the other paths significantly by 0.1), it trivially cannot be part of an optimal path pair since the following. Take the pair of paths P_2 and $P_1 = \{e, f\}$. Then, since $\mathrm{FP}(e, g) = 0.2$, together with $e \in P_1$, and $g \in P_2$ means that the probability that t is available from s on at



Path	Availability of path	Path pair	Availability of path pair
$P_1 := \{e, f\}$	0.5	$P_1 + P_2$	0.8
$\mathbf{P_2} := \{\mathbf{g}, \mathbf{h}\}$	**0.6**	$\mathbf{P_1} + \mathbf{P_3}$	**1**
$P_3 := \{i, j\}$	0.5	$P_2 + P_3$	0.8

Fig. 1: A toy example on the input graph G, related failure probabilities stored in either FP[G] or CFP[G]. The table included depicts the availabilities of each st-path and each st-path pair. Although path P_2 is the safest, it is not part of any safest path pair.

least one of P_1 and P_2 is upper bounded by $1 - \text{FP}(e, g) = 0.8$. In fact, since there is no other failure state that intersects both P_1 and P_2, $A_{st}(P_1, P_2) = 0.8$. For similar reasons, we also have $A_{st}(P_2, P_3) = 0.8$. On the other hand, though, there is no failure state in this example that would cut both P_1 and P_3. This means that, after the next single disaster event, there exists a connection between s and t through either P_1 or P_3 with probability 1.

2.2 $\mathcal{NP}$-hardness

Theorem 1. *Problems* `Safest Path Pair FP` *and* `Safest Path Pair CFP` *(Problems 1 and 2) are $\mathcal{NP}$-hard.*

Proof. To prove the $\mathcal{NP}$-hardness of the problems, we use a reduction to the safest (single) st-path problems as defined in [28] (using either FP[G] or CFP[G] as input). Our reduction is very straightforward, and it is intuitively depicted in Fig. 2. To put it briefly, the degree of both s and t is 2, s' and s'' being the neighbors of s, while t' and t'' being the neighbors of t. Intuitively, we consider that between s' and t', there is a safest (single) $s't'$-path problem instance as defined in [28]; and between s'' and t'' we consider the same safest path instance I. More precisely, if in the single safest path problem instance there is a link set with $\text{FP}(S) > 0$, then, in the transformed problem version, we set $\text{FP}(S' \cup S'') := \text{FP}(S)$, where S' and S'' are the image of S in the problem instances between pairs $s't'$ and $s''t''$, respectively. We can define the reduction using CFP similarly. Then, based on our construction and the proof of [28, Thm 2.4], it is $\mathcal{NP}$-hard to decide whether there exists an st-path pair that has an availability of at least $1/2$. The proof follows. □

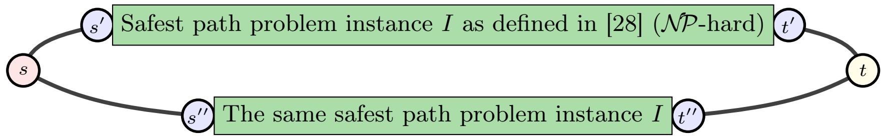


Fig. 2: Intuitive illustration for the proof of $\mathcal{NP}$-hardness.

We note that the construction used by the above proof underpins the intuition that computing a safest st-path *pair* is at least as hard as computing a *single* safest st-path.

3 Safest Path Pair Ant Colony Optimization Algorithm - SPP-ACO

The ant colony optimization algorithm - first proposed in [8] - is a powerful optimization tool for graph-based computational problems, based on the metaphor of the ants behavior: ants communicate indirectly based on the pheromones they leave behind them. An ACO algorithm finds the optimal solutions by the virtue of the aforementioned pheromones. Initially, the ants only have a local understanding of the problem provided by the heuristic information, but over time, due to an aspect of randomness, they will find alternative paths. If any alternative paths prove to be better, then the pheromones laid will be stronger, more ants will discover that route, thus the algorithm gains a better global understanding of the problem. In this article, we adopted the Max-Min Ant System [24] algorithm, which controls the maximum and minimum pheromone level on trails.

In this section, we present the basic elements of the ACO algorithm: heuristic information, pheromone setting, and solution generation. In our algorithm, the heuristic information is defined as the negative logarithm of the failure probability associated with a given edge, considering only individual edge probabilities. If an edge does not have an explicitly assigned probability, we assign it a small default value $\epsilon_{\text{prob}} > 0$, as the logarithm of zero is undefined. This value is selected to be significantly smaller than any failure probability in the original set $\text{CFP}[G]$; specifically, we used $\epsilon_{\text{prob}} = 10^{-8}$ in our experiments. Furthermore, if two nodes are not directly connected (i.e., they are not neighbors), the heuristic information is set to zero to ensure that such edges are not considered by the ants. This behavior is formalized in Eq. (1), where $\mathcal{N}_i$ denotes the set of neighbors of node i, and $e_{i,j}$ represents the edge connecting nodes i and j.

$$\eta_{ij} = \begin{cases} -\log \text{CFP}(\{e_{i,j}\}), & \text{if } j \in \mathcal{N}_i \\ 0, & \text{else.} \end{cases} \tag{1}$$

The pheromone limitations ($\tau_{\max}$ and $\tau_{\max}$) are chosen based on the following formulas:

$$\tau_{\max} = f_{gb}/(1-\rho), \tau_{\min} = \epsilon \cdot \tau_{\max}, \tag{2}$$

where f_{gb} is the fitness of the best solution, ρ is the evaporation coefficient, and ϵ denotes the pheromone proportion coefficient.

The combination of pheromone placement and evaporation is described in Eq. (3), where τ_{ij} is the pheromone level between nodes i and j:

$$\tau_{i,j}^{(t)} = \begin{cases} (1-\rho)\cdot\tau_{ij}^{(t-1)}, & \text{if nodes } i,j \text{ are not on the best route.} \\ (1-\rho)\cdot\tau_{ij}^{A(t-1)} - \log\left(1-A(P)\right), & \text{if nodes } i,j \text{ are on the best route.} \end{cases} \tag{3}$$

The transition probabilities guiding the ant's movement are computed as described in Eq. (4), where τ_{ij} represents the pheromone level on the edge between nodes i and j, η_{ij} denotes the heuristic information for the same edge, and $\mathcal{N}_i$ is the set of neighbors of node i.

$$p(i,j) = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{v_q\in\mathcal{N}_i}(\tau_{iq})^{\alpha}(\eta_{iq})^{\beta}}\ \text{, if } v_j \in \mathcal{N}_i \tag{4}$$

The SPP-ACO algorithm works as follows. First, we duplicate each node and edge into a second component in the graph and connect the two components through the original target node and its copy. Then we set the copy of the starting node as the new target. After this, all ants are placed in the starting point s. Based on the heuristic information and pheromone level, they choose the next node. Nodes can be visited at most once; the last visited node must be t. The algorithm iteratively executes the following steps until the stopping criterion—defined as reaching the maximum number of iterations—is met. In each iteration, the global and iteration-best paths are determined. Based on these, new pheromone limits are computed, and on the best path, pheromone level is increased, while pheromone on edges not included in this path evaporates at a constant rate. This process is summarized in Algorithm 1.

Regarding the path generation method, if an ant cannot reach the endpoint, no path will be returned. The path generation algorithm is detailed in [28].

After the algorithm finds a solution, in order to assess its fitness, it has to transform the path on the duplicated graph back into a path pair on the original graph. Since the duplicated part of the graph corresponds one-to-one to the original part just with offset indices, it is enough to leave out the edge that connects the two components and subtract $|V|$ from each node in the second part of the path to get to a path on the original graph, while retaining the first part as the other.

4 Numerical experiments

4.1 Simulation settings

Benchmarks For benchmarks, we use four real-world settings[3]: the 22_optic network [26] has 22 nodes and 45 edges, the Italy network (a.k.a. interroute_v2)

[3] https://github.com/jtapolcai/regional-srlg/blob/master/psrlg/JSACdata.zip

Algorithm 1: Shortest Path Pair-Ant Colony Optimization (SPP-ACO)

Input: Graph $G = (V, A)$, cumulative failure probabilities CFP[G], ACO parameters: $\alpha, \beta, \rho, \epsilon$, nrOfAnts; nrOfIterations, nodes s and t
Output: A safest st-path pair
Duplicate graph G into G'
Connect G and G' through t and t'
$t := s'$
Initialize pheromone trails
$i := 0$
while $i <$ *nrOfIterations* **do**
  $S := \emptyset$
  repeat
    Construct a new path P from s to t
    $S := S \cup \{P\}$
  until $|S| =$ *nrOfAnts*;
  Calculate the iteration-best and global-best paths: $P_{\mathtt{ib}}$ and $P_{\mathtt{best}}$, respectively
  Compute pheromone trail limits ($\tau_{\mathtt{min}}$, $\tau_{\mathtt{max}}$) based on Eq. (2)
  Update pheromone trail on $P_{\mathtt{ib}}$ based on Eq. (3)
  $i := i + 1$
return P_{best}

[29] contains 25 nodes and 34 edges, the cost266 network has 37 nodes and 57 edges, while the janos_us network [26] has 26 nodes and 42 edges. The failure data used for numerical experiments for each network is taken from [30].

Parameter tuning To test the proposed SP-ACO algorithm[4], we run a parameter test for the following four parameters: $\alpha \in \{0.5, 1, 1.5\}$, $\beta \in \{0.5, 1, 1.5\}$, $\epsilon \in \{0.1, 0.3\}$, and $\rho \in \{0.1, 0.3\}$. The number of ants was fixed to 25, and the total number of iterations was set to 50. The parameter tuning was performed on a real-world network, the Italy graph with intensity tolerance 6.[5] Totally 36 combinations of parameters were analyzed; but due to the sheer amount of data, only some representative cases have their numerical results presented in Table 1. To get a comprehensive picture of the results, we used two methods: Wilcoxon tests to assess whether the samples differ in a statistically significant way and a chess tournament comparison to rank the configurations.

Twenty independent runs were conducted; mean values, standard deviation, and maximum values are reported. Based on the numerical results, there were multiple configurations with similar results, but in the end configuration 21 was selected for further experiments, which means $\alpha = 1.5$, $\beta = 0.5$, $\rho = 0.1$, $\epsilon = 0.3$.

[4] Available at: https://github.com/VaranTavers/safest_route_pair_jl.
[5] Measured according to the Mercalli-Cancani-Sieberg (MCS) scale [22] that is used in Italy to measure the intensity of shaking at any given location due to an earthquake. The MCS scale ranges from 1 to 12. An intensity of ≥ 6 may cause structural damage.

Table 1: Results of parameter tuning (20 independent runs) on some node-pairs of the italy graph (intensity tolerance 6) containing the mean, standard deviation (out of 50 generations).

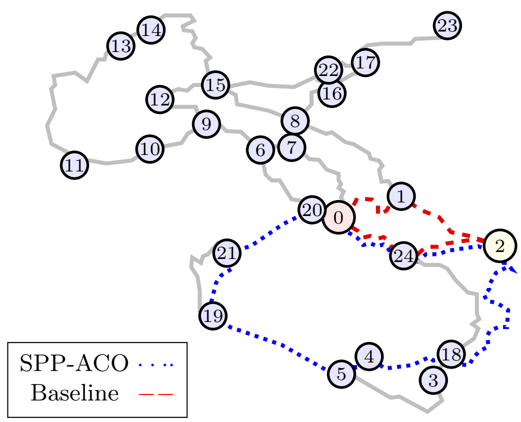
nr	α	β	ρ	ϵ	italy (0,7)	italy (1, 17)	italy (2, 14)	italy (3, 15)	italy (18, 23)
1	0.5	0.5	0.1	0.1	5.772 ± 0	5.756 ± 0.045	6.268 ± 0.045	5.218 ± 0.013	4.524 ± 0.011
2	1	0.5	0.1	0.1	5.772 ± 0	5.788 ± 0	6.299 ± 0	5.21 ± 0.069	4.53 ± 0
3	1.5	0.5	0.1	0.1	5.772 ± 0	5.723 ± 0.168	6.299 ± 0	5.164 ± 0.126	4.527 ± 0.009
4	0.5	1	0.1	0.1	5.772 ± 0	5.748 ± 0.045	6.273 ± 0.041	5.221 ± 0.011	4.523 ± 0.009
5	1	1	0.1	0.1	5.772 ± 0	5.788 ± 0	6.299 ± 0	5.226 ± 0	4.529 ± 0.007
6	1.5	1	0.1	0.1	5.772 ± 0	5.772 ± 0.039	6.299 ± 0	5.226 ± 0	4.526 ± 0.011
7	0.5	1.5	0.1	0.1	5.772 ± 0	5.756 ± 0.049	6.285 ± 0.035	5.219 ± 0.016	4.516 ± 0.012
8	1	1.5	0.1	0.1	5.772 ± 0	5.783 ± 0.024	6.299 ± 0	5.226 ± 0	4.527 ± 0.009
9	1.5	1.5	0.1	0.1	5.772 ± 0	5.736 ± 0.119	6.299 ± 0	5.226 ± 0	4.519 ± 0.038
10	0.5	0.5	0.3	0.1	5.772 ± 0	5.746 ± 0.049	6.285 ± 0.034	5.22 ± 0.015	4.525 ± 0.01
11	1	0.5	0.3	0.1	5.772 ± 0	5.788 ± 0	6.299 ± 0	5.226 ± 0	4.527 ± 0.009
12	1.5	0.5	0.3	0.1	5.772 ± 0	5.739 ± 0.131	6.299 ± 0	5.164 ± 0.126	4.521 ± 0.023
13	0.5	1	0.3	0.1	5.772 ± 0	5.757 ± 0.046	6.291 ± 0.024	5.22 ± 0.012	4.524 ± 0.009
14	1	1	0.3	0.1	5.772 ± 0	5.783 ± 0.024	6.299 ± 0	5.226 ± 0	4.529 ± 0.007
15	1.5	1	0.3	0.1	5.772 ± 0	5.704 ± 0.175	6.299 ± 0	5.195 ± 0.094	4.523 ± 0.013
16	0.5	1.5	0.3	0.1	5.772 ± 0	5.773 ± 0.03	6.295 ± 0.018	5.223 ± 0.007	4.526 ± 0.008
17	1	1.5	0.3	0.1	5.772 ± 0	5.783 ± 0.024	6.299 ± 0	5.226 ± 0	4.526 ± 0.011
18	1.5	1.5	0.3	0.1	5.772 ± 0	5.733 ± 0.132	6.299 ± 0	5.19 ± 0.096	4.485 ± 0.071
19	0.5	0.5	0.1	0.3	5.772 ± 0	5.682 ± 0.052	6.204 ± 0.077	5.194 ± 0.027	4.512 ± 0.017
20	1	0.5	0.1	0.3	5.772 ± 0	5.767 ± 0.041	6.287 ± 0.029	5.215 ± 0.015	4.525 ± 0.006
21	1.5	0.5	0.1	0.3	5.772 ± 0	5.788 ± 0	6.299 ± 0	5.226 ± 0	4.53 ± 0
22	0.5	1	0.1	0.3	5.772 ± 0	5.692 ± 0.055	6.207 ± 0.118	5.198 ± 0.028	4.501 ± 0.026
23	1	1	0.1	0.3	5.772 ± 0	5.764 ± 0.042	6.287 ± 0.029	5.223 ± 0.007	4.523 ± 0.009
24	1.5	1	0.1	0.3	5.772 ± 0	5.788 ± 0	6.299 ± 0	5.226 ± 0	4.53 ± 0
25	0.5	1.5	0.1	0.3	5.772 ± 0	5.73 ± 0.052	6.225 ± 0.064	5.194 ± 0.021	4.507 ± 0.018
26	1	1.5	0.1	0.3	5.772 ± 0	5.771 ± 0.037	6.291 ± 0.025	5.22 ± 0.01	4.523 ± 0.01
27	1.5	1.5	0.1	0.3	5.772 ± 0	5.788 ± 0	6.299 ± 0	5.226 ± 0	4.53 ± 0
28	0.5	0.5	0.3	0.3	5.772 ± 0	5.682 ± 0.044	6.168 ± 0.11	5.194 ± 0.026	4.51 ± 0.02
29	1	0.5	0.3	0.3	5.772 ± 0	5.756 ± 0.046	6.283 ± 0.032	5.219 ± 0.012	4.527 ± 0.006
30	1.5	0.5	0.3	0.3	5.772 ± 0	5.783 ± 0.024	6.299 ± 0	5.226 ± 0	4.53 ± 0
31	0.5	1	0.3	0.3	5.772 ± 0	5.724 ± 0.09	6.199 ± 0.087	5.204 ± 0.018	4.498 ± 0.034
32	1	1	0.3	0.3	5.772 ± 0	5.764 ± 0.046	6.287 ± 0.029	5.223 ± 0.007	4.528 ± 0.005
33	1.5	1	0.3	0.3	5.772 ± 0	5.783 ± 0.024	6.299 ± 0	5.226 ± 0	4.527 ± 0.009
34	0.5	1.5	0.3	0.3	5.772 ± 0	5.72 ± 0.061	6.22 ± 0.102	5.204 ± 0.018	4.5 ± 0.017
35	1	1.5	0.3	0.3	5.772 ± 0	5.76 ± 0.042	6.293 ± 0.026	5.222 ± 0.01	4.527 ± 0.006
36	1.5	1.5	0.3	0.3	5.772 ± 0	5.782 ± 0.024	6.299 ± 0	5.226 ± 0	4.526 ± 0.011

Comparisons with other methods To evaluate the relative performance of the proposed algorithm, we compare it with two algorithms: a strawman approach and a simple genetic algorithm.

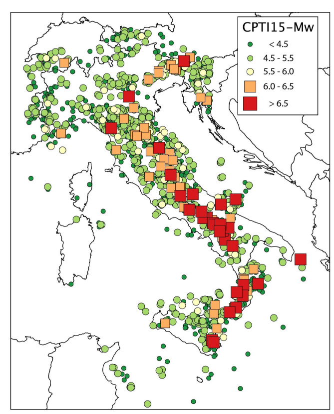
For a given source-target pair s and t, the strawman (baseline) algorithm first computes a cheapest st-path P_1 using Dijkstra's algorithm, where for each

Table 2: Results of the SPP-ACO on all benchmark networks using two intensity tolerances (it6 and it7) compared to the baseline implementation. First three numerical columns indicate that mean values how many times where worst (<), the same (=) or better (>) than the baseline algorithm, while next three column values indicate the numbers of the best result obtained in 20 independent runs were worse (<), the same (=) or better (>) than the baseline algorithm.

Network	mean			max		
	$A(P_{\text{SPP-ACO}}) < A(P_{\text{baseline}})$	$A(P_{\text{SPP-ACO}}) = A(P_{\text{baseline}})$	$A(P_{\text{SPP-ACO}}) > A(P_{\text{baseline}})$	$A(P_{\text{SPP-ACO}}) < A(P_{\text{baseline}})$	$A(P_{\text{SPP-ACO}}) = A(P_{\text{baseline}})$	$A(P_{\text{SPP-ACO}}) > A(P_{\text{baseline}})$
22_optic_it6	30	15	186	0	45	186
22_optic_it7	13	22	196	0	26	205
italy_it6	1	87	212	0	88	212
italy_it7	6	97	197	0	103	197
cost266_it6	36	31	599	0	59	607
cost266_it7	63	64	539	1	125	540
janos_us_it6	17	55	253	0	72	253
janos_us_it7	20	37	268	0	57	268



(a) Path pairs generated by SPP-ACO (dotted), and the baseline (dashed) between Rome (id 0) and Bari (id 2).



(b) Historical earthquakes from Italian catalog [20]. M_w = moment magnitude.

Fig. 3: Based on the seismic hazard data distilled from an Italian historical catalog, the SPP-ACO detours one of the paths between the nodes of Rome (id 0) and Bari (id 2) to cross Sardinia and Sicily. Intuitively, this maximizes the probability that at least one of the paths remains intact when an earthquake strikes between Rome and Bari.

Table 3: Results of the SPP-ACO on all benchmark networks using one intensity tolerance (it6) compared to the SPP-GA. First three numerical columns indicate that mean values how many times where worst (<), the same (=) or better (>) than the mean of the GA algorithm, while next three column values indicate the numbers of the best result obtained in 20 independent runs were worse (<), the same (=) or better (>) than the SPP-GA.

Network	mean			max		
	$A\left(P_{\text{SPP-ACO}}\right) < A\left(P_{\text{SPP-GA}}\right)$	$A\left(P_{\text{SPP-ACO}}\right) = A\left(P_{\text{SPP-GA}}\right)$	$A\left(P_{\text{SPP-ACO}}\right) > A\left(P_{\text{SPP-GA}}\right)$	$A\left(P_{\text{SPP-ACO}}\right) < A\left(P_{\text{SPP-GA}}\right)$	$A\left(P_{\text{SPP-ACO}}\right) = A\left(P_{\text{SPP-GA}}\right)$	$A\left(P_{\text{SPP-ACO}}\right) > A\left(P_{\text{SPP-GA}}\right)$
22_optic_it6	131	58	42	1	231	0
22_optic_it7	132	63	37	0	232	0
italy_it6	4	99	207	0	297	3
italy_it7	3	105	202	0	299	1
cost266_it6	210	122	334	45	661	10
cost266_it7	337	143	186	63	603	0
janos_us_it6	39	85	201	3	314	8
janos_us_it7	59	81	185	6	312	7

edge e, its cost $c(e)$ is set to CFP(e). Then, a second st-path P_2 is calculated, also using a Dijkstra, but with a modified cost function c', where for each edge e that is not part of P_1, $c'(e) := c(e)$, while for $e \in P_1$, $c'(e) := c(e) + 1$. The $+1$ ensures that this edge will not be chosen unless absolutely necessary. In this way, each of the resulting paths P_1 and P_2 is reasonably safe, and as a tuple, they have 'few' edges in common, making the output of this approach a reasonable naive solution to the problem.

The genetic algorithm (SPP-GA) assigns integer codes to the edges of each FP with three possible values: 1- allowed to be part of the first path, 2- allowed to be part of the second path, 0- not part of any path. The fitness function is the same as for the ACO algorithm. Regarding the used operators uniform crossover is used and random resetting mutation. For parent selection roulette wheel selection is used and the survivals are selected with elitist selection. The parameters used for this algorithm are the following: 100 generations, population size is 50, mutation rate of 0.3, and cross-over rate of 0.9.

Results and discussion Tables 2 and 3 present the obtained results for the benchmark real-world networks. For each network, two variants were considered with two different values of intensity tolerance. For the SPP-ACO algorithm, 20 independent runs were conducted for each node pair st. Considering the mean values, our proposed algorithm outperformed the baseline method in most of the cases: from a total of 3044 tests, the SPP-ACO outperformed the baseline method in 80.59% of the total cases, and in 16.65% the same results were obtained. Regarding the comparisons with the SPP-GA, our algorithm outperformed the SPP-GA in (considering the mean values) in 45.79% of the total cases, and in

24.83% the same results were obtained. Figure 3 presents visually the detected path pairs between Rome and Bari.

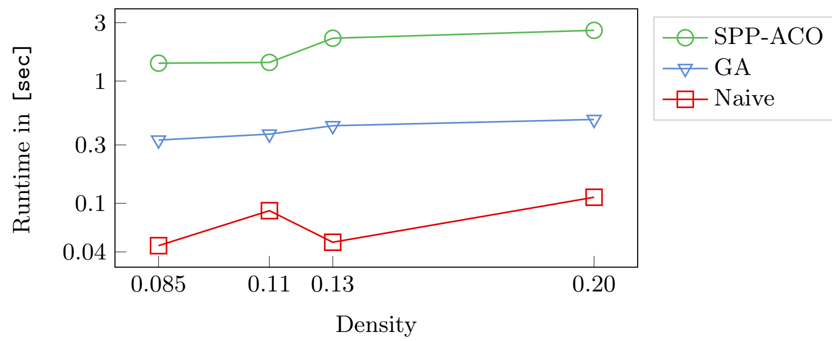


Fig. 4: Runtimes of the three algorithms presented in `[sec]` plotted against the density in the tested graphs (with intensity tolerance 6) for a single *st* pair, averaging 20 runs.

Runtime analysis All three algorithms were implemented in Julia, therefore a runtime comparison was straightforward. The times were measured for all four networks between a randomly chosen *st* pair, 20 independent runs were conducted. The results are presented in Figure 4. As a general conclusion, it can be clearly established that the SPP-ACO algorithm is slower than the greedy-based baseline method and the genetic algorithm, but this behavior aligns with the general expectation regarding the runtime differences between nature-inspired and greedy algorithms.

5 Conclusions and future work

As proved above, the safest path pair computation problem, where multiple correlated failures can appear, is a challenging $\mathcal{NP}$-hard optimization problem with several application possibilities, for example, in determining vulnerable network parts. In this article, we propose the Safest Path Pair Ant Colony Optimization algorithm (SPP-ACO) to solve the problem. To guide the ants the network is extended, and problem-specific information is introduced in the heuristic function. Numerical experiments were conducted on real-world problems, and comparisons with a baseline method were performed. Results prove the effectiveness of the proposed approach.

Future work will address the adaptation of other nature-inspired algorithms for the proposed problem, parallelization of the proposed algorithm, and the study of hybrid variants of the algorithm. From a problem-specific point of view further work will address other variants of the problem, for example cascading failures.

Acknowledgments. This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS-UEFISCDI, project number PN-IV-P2-2.1-TE-2023-1977.

References

1. Aceto, G., Botta, A., Marchetta, P., Persico, V., Pescapé, A.: A comprehensive survey on internet outages. Journal of Network and Computer Applications **113**, 36–63 (2018)
2. Bellman, R.: On a routing problem. Quarterly of applied mathematics **16**(1), 87–90 (1958)
3. Bérczi-Kovács, E., Gyimesi, P., Vass, B., Tapolcai, J.: DateLine: efficient algorithm for computing region disjoint paths in backbone networks. IEEE JSAC (2025)
4. Bienstock, D.: Some generalized max-flow min-cut problems in the plane. Mathematics of Operations Research **16**(2), 310–333 (1991)
5. Cui, W., Stoica, I., Katz, R.H.: Backup path allocation based on a correlated link failure probability model in overlay networks. In: 10th IEEE International Conference on Network Protocols, 2002. Proceedings. pp. 236–245. IEEE (2002)
6. Di Caprio, D., Ebrahimnejad, A., Alrezaamiri, H., Santos-Arteaga, F.J.: A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights. Alexandria Engineering Journal **61**(5), 3403–3415 (2022)
7. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
8. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE computational intelligence magazine **1**(4), 28–39 (2006)
9. Fister Jr, I., Yang, X.S., Fister, I., Brest, J., Fister, D.: A brief review of nature-inspired algorithms for optimization. arXiv preprint arXiv:1307.4186 (2013)
10. General, M., Pezoa, J.E.: Reliable path provisioning in the presence of multiple correlated failures. In: 2016 IEEE 37th Sarnoff Symposium. pp. 1–2. IEEE (2016)
11. Horoba, C., Sudholt, D.: Ant colony optimization for stochastic shortest path problems. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation. pp. 1465–1472 (2010)
12. Lee, H.W., Modiano, E., Lee, K.: Diverse routing in networks with probabilistic failures. IEEE/ACM Trans. Netw. **18**(6), 1895–1907 (2010)
13. Li, Y., Zhang, D., Tan, K.L.: Real-time targeted influence maximization for online advertisements (2015)
14. Liu, J., Zhang, J., Zhao, Y., Ma, C., Yang, H., Li, W., Xin, J., Chen, B.: Differentiated quality-of-protection provisioning with probabilistic SRLG in flexi-grid optical networks. In: OSA Asia Communications and Photonics Conference. pp. AF2G–8 (2013)
15. Ok, S.H., Seo, W.J., Ahn, J.H., Kang, S., Moon, B.: An ant colony optimization approach for the preference-based shortest path search. Journal of the Chinese Institute of Engineers **34**(2), 181–196 (2011)
16. Oostenbrink, J., Kuipers, F.: Computing the impact of disasters on networks. ACM SIGMETRICS Performance Evaluation Review **45**(2), 107–110 (2017)
17. Ou, C.S., Mukherjee, B.: Survivable Optical WDM Networks. Springer Science & Business Media (2005)
18. Pašić, L., Pašić, A., Mogyorósi, F., Pašić, A.: FRADIR meets availability. In: 2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020. pp. 1–6. IEEE (2020)

19. Pašić, A., ao Silva, R.G., Mogyorósi, F., Vass, B., Gomes, T., Babarczi, P., Revisnyei, P., Tapolcai, J., Rak, J.: eFRADIR: An Enhanced FRAmework for DIsaster Resilience. IEEE Access **9**, 13125–13148 (2021)
20. Rovida, A., Locati, M., Camassi, R., Lolli, B., Gasperini, P.: Cpti15, the 2015 version of the parametric catalogue of italian earthquakes. Istituto Nazionale di Geofisica e Vulcanologia (2016)
21. Sheng, M., Li, J., Shi, Y.: Critical nodes detection in mobile ad hoc network. In: 20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06). vol. 2, pp. 5–pp. IEEE (2006)
22. Sieberg, A.: Erdebeben. Handbuch der Geophysic **4**, 552–554 (1931)
23. Somani, A.: Survivability and traffic grooming in WDM optical networks. Cambridge University Press (2006)
24. Stützle, T., Hoos, H.H.: Max–min ant system. Future Generation Computer Systems **16**(8), 889–914 (2000)
25. Suurballe, J.W., Tarjan, R.E.: A quick method for finding shortest pairs of disjoint paths. Networks **14**(2), 325–336 (1984)
26. Tapolcai, J., Rónyai, L., Vass, B., Gyimóthi, L.: List of shared risk link groups representing regional failures with limited size. In: IEEE INFOCOM. Atlanta, USA (May 2017)
27. Tapolcai, J., Vass, B., Heszberger, Z., Biró, J., Hay, D., Kuipers, F.A., Rónyai, L.: A tractable stochastic model of correlated link failures caused by disasters. In: Proc. IEEE INFOCOM. Honolulu, USA (Apr 2018)
28. Tasnádi, Z., Vass, B., Gaskó, N.: Ant colony optimization algorithm for safest path computation in presence of correlated failures in backbone networks. In: ACM Genetic and Evolutionary Computation Conference (GECCO). Málaga, Spain (Jul 2025)
29. Valentini, A., Vass, B., Oostenbrink, J., Csák, L., Kuipers, F., Pace, B., Hay, D., Tapolcai, J.: Network resiliency against earthquakes. In: 2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM). pp. 1–7 (Oct 2019)
30. Vass, B., Tapolcai, J., Heszberger, Z., Bíró, J., Hay, D., Kuipers, F.A., Oostenbrink, J., Valentini, A., Rónyai, L.: Probabilistic shared risk link groups modelling correlated resource failures caused by disasters. IEEE Journal on Selected Areas in Communications (JSAC) - issue on Latest Advances in Optical Networks for 5G Communications and Beyond (2021)
31. Vass, B., Tapolcai, J., Hay, D., Oostenbrink, J., Kuipers, F.: How to model and enumerate geographically correlated failure events in communication networks. In: Guide to Disaster-Resilient Communication Networks, pp. 87–115. Springer (2020)
32. Vass, B., Revisnyei, P., Pašić, A.: Computing safest st-paths in backbone networks: Efficiently solvable cases and fast heuristics. In: 2024 14th International Workshop on Resilient Networks Design and Modeling (RNDM)
33. Yang, S., Trajanovski, S., Kuipers, F.A.: Shortest paths in networks with correlated link weights. In: 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 660–665. IEEE (2015)
34. Zhou, D., Subramaniam, S.: Survivability in optical networks. IEEE network **14**(6), 16–23 (2000)