

## ACCEPTED VERSION

Aneta Neumann, Sharlotte Gounder, Xiankun Yan, Gregory Sherman, Benjamin Campbell, Mingyu Guo, Frank Neumann

### **Diversity Optimization for the Detection and Concealment of Spatially Defined Communication Networks**

Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '23), 2023 / Paquete, L. (ed./s), pp.1436-1444

© 2023 by the Association for Computing Machinery, Inc. (ACM).

Definitive Version of Record: <http://dx.doi.org/10.1145/3583131.3590405>

#### PERMISSIONS

<https://authors.acm.org/author-services/author-rights>

### **ACM Author Rights**

#### **Post**

Otherwise known as "Self-Archiving" or "Posting Rights", all ACM published authors of magazine articles, journal articles, and conference papers retain the right to post the pre-submitted (also known as "pre-prints"), submitted, accepted, and peer-reviewed versions of their work in any and all of the following sites:

- Author's Homepage
- Author's Institutional Repository
- Any Repository legally mandated by the agency or funder funding the research on which the work is based
- Any Non-Commercial Repository or Aggregation that does not duplicate ACM tables of contents. Non-Commercial Repositories are defined as Repositories owned by non-profit organizations that do not charge a fee to access deposited articles and that do not sell advertising or otherwise profit from serving scholarly articles.

**11 April 2024**

<http://hdl.handle.net/2440/139310>

# Diversity Optimization for the Detection and Concealment of Spatially Defined Communication Networks

Aneta Neumann  
Optimisation and Logistics  
The University of Adelaide  
Adelaide, Australia

Sharlotte Gounder  
Optimisation and Logistics  
The University of Adelaide  
Adelaide, Australia

Xiankun Yan  
Optimisation and Logistics  
The University of Adelaide  
Adelaide, Australia

Gregory Sherman  
Defence Science and  
Technology Group  
Edinburgh  
Edinburgh, Australia

Benjamin Campbell  
Defence Science and  
Technology Group  
Edinburgh  
Edinburgh, Australia

Mingyu Guo  
Optimisation and Logistics  
The University of Adelaide  
Adelaide, Australia

Frank Neumann  
Optimisation and Logistics  
The University of Adelaide  
Adelaide, Australia

## ABSTRACT

In recent years, computing diverse sets of high quality solutions for an optimization problem has become an important topic. The goal of computing diverse sets of high quality solutions is to provide a variety of options to decision makers, allowing them to choose the best solution for their particular problem. We consider the problem of constructing a wireless communication network for a given set of entities. Our goal is to minimize the area covered by the senders' transmissions while also avoiding adversaries that may observe the communication. We provide evolutionary diversity optimization (EDO) algorithms for this problem. We provide a formulation based on minimum spanning forests that are used as a representation and show how this formulation can be turned into a wireless communication network that avoids a given set of adversaries. We evaluate our EDO approach based on a number of benchmark instances and compare the diversity of the obtained populations in respect to the quality criterion of the given solutions as well as the chosen algorithm parameters. Our results demonstrate the effectiveness of our EDO approaches for the detection and concealment of communication networks both in terms of the quality and the diversity of the obtained solutions.

## KEYWORDS

Evolutionary Diversity Optimization, Quality Diversity, Minimum Area Spanning Tree, Low Probability Detection

### ACM Reference Format:

Aneta Neumann, Sharlotte Gounder, Xiankun Yan, Gregory Sherman, Benjamin Campbell, Mingyu Guo, and Frank Neumann. 2023. Diversity Optimization for the Detection and Concealment of Spatially Defined Communication Networks. In *Genetic and Evolutionary Computation Conference (GECCO '23)*, July 15–19, 2023, Lisbon, Portugal. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

GECCO '23, July 15–19, 2023, Lisbon, Portugal

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Evolutionary algorithms (EAs) are state-of-the-art, efficient heuristic search methods that solve complex optimization problems by employing the Darwinian theory of natural selection. Due to their flexibility to capture global solutions and ability to self adapt, evolutionary algorithms have been applied to various real-world problems in areas such as engineering [2, 6], supply chain management [7, 9] and communication networks [15, 18].

Computing diverse sets of solutions for a given problem has recently gained significant attention in the evolutionary computation literature under the terms evolutionary diversity optimization (EDO) [11, 13, 20–24, 28] and quality diversity (QD) [17, 19, 25]. QD algorithms usually work with a variable population size and try to capture the best possible solutions within different niches of a given behaviour space. On the contrary, EDO algorithms work with a fixed population size and try to make this population as diverse as possible according to a given diversity measure under the condition that all solutions in the population meet a given quality criterion.

### 1.1 Background: Low probability detection problem

In recent years, the Low Probability of Detection (LPD) problem has gained significant attention in the mobile networks and data privacy/security community. The LPD problem [29] is thought of as a binary detection problem, since the detector is interested in detecting the transmission itself rather than obtaining what was communicated between the parties. Multi-hop communication is essential for communication required over long distances, [29] explains that single-hop communication is often not a realistic solution for long distances as it requires a higher transmission power which can be easily detected by nearby adversaries. It was highlighted in [4] that reducing the transmission power will decrease the likelihood of the detection, however, as transmission ranges overlap, this results in a nonlinear relationship between transmission power and detection probability. Therefore, the detectability footprint is a more suitable variable to optimize to achieve LPD, as the detectability footprint is the *area in which the radiated power is above a detectable threshold* [4].

LPD is concerned with the technological development that enables the successful detection and concealment of certain activities or objects in wireless systems and networks for all voice and data communication. The basic LPD concept consists of a transmitter that aims to communicate with a receiver, while a detector attempts to identify if a transmission has occurred. Campbell et al. [4] minimized the detectability footprint in order to solve the LPD problem by generating a spanning tree that maintains a minimum detectability footprint and ensures effective transmission.

The minimum area spanning tree (MAST) problem [5] is a tree-based network topology problem used to reduce the probability of detection by an adversary. The problem seeks to connect nodes in a network in a way that minimizes the amount of physical area occupied by the network. This makes it more difficult for an adversary to detect the network due to the reduced physical footprint. Solving the problem is useful in phone networks, wireless local area networks, satellite communication networks, and other security-sensitive industrial applications where maintaining a low profile is important. Previous approaches developed in [4] use a brute force technique and were only able to tackle graphs with up to 25 nodes.

## 1.2 Our contribution

In this paper, we develop evolutionary algorithms for a variant of the MAST problem that takes into account a given set of adversaries. In order to tackle a given set of adversaries, we formulate the problem as the adversarial minimum area spanning forest (AMASF) problem which takes into account communication connections that can not be used due to the Euclidean proximity of adversaries. We tackle the AMASF problem both in terms of classical optimization and in terms of diversity optimization. We first develop evolutionary algorithms for this problem in terms of its optimization variant. Our goal here is to minimize the area covered by the senders under the condition that maximal communication across the network is possible without allowing the adversaries to observe the communication. Afterwards, we consider the construction of diverse sets of solutions using the EDO approach. This allows decision makers to select the solutions that should be implemented from a given diverse set of high quality solutions. It also shows the decision maker the different options of establishing the communication network and thereby helps to identify bottlenecks and critical areas of the network.

We investigate our algorithms on different instances with up to 200 nodes in the Euclidean plane. Our results show the amount of diversity that can be obtained in the population depended on the quality threshold that is used as part of the EDO approach. Furthermore, we provide results for different parent population sizes which amount to the different number of solutions provided to the decision maker. We showcase final populations for some of our example instances in order to provide insights into the differences in the solutions provided by our EDO approaches.

In Section 2 and 3, we define the problem statement and introduce spanning trees as representations of possible solutions for the Minimum Area Spanning Tree problem. We describe a scenario of the aforementioned problem where the adversaries are present in Section 4. We provide the definition of evolutionary diversity

optimization and introduce our approach for the detection and concealment of spatially defined communication networks, as well as new mutation operators, in Section 5. We present and discuss our experimental results of our new approach in Section 6. Finally, we finish with some concluding remarks.

## 2 PROBLEM STATEMENT

Let  $G = (V, E)$  be a directed connected graph with node set  $V$ , edge set  $E$  and distance function  $d : E \rightarrow R^+$  which assigns positive distance between each edge. We denote by  $n = |V|$ ,  $m = |E|$  the size of the node and edge set respectively.

We consider the complete graph  $G = (V, E)$  in the Euclidean plane where the node  $v_i \in V$  has position  $p_i = (x_i, y_i)$ ,  $1 \leq i \leq n$ . So that the distances  $d$  are given as the Euclidean distances

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

between a pair of nodes. Note that the resulting graph is symmetric.

Given  $n$  vertices  $v_i$  with locations  $v_i = (x_i, y_i)$ ,  $1 \leq i \leq n$ , in the Euclidean plane, the goal is to determine a set of circles  $C = \{C_1, \dots, C_n\}$  where circles  $C_i$  has centre  $v_i$  and radius  $r_i$  such that the resulting graph is connected and the area covered by the circles is minimal.

Formally, a solution is given as a set  $C = \{C_1, \dots, C_n\}$ . The circle  $C_i$  is defined as  $C_i = (v_i, r_i)$ , where  $v_i = (x_i, y_i)$  is the centre of the circle and  $r_i$  is the radius. For a given set of circles  $C = C_1, \dots, C_n$  there is an edge from  $(v_i, v_j)$  in the connectivity graph  $G(C) = (V, E(C))$  if  $d(v_i, v_j) \leq r_i + r_j$ . Here an edge  $(v_i, v_j)$  indicates a direct communication from  $v_i$  to  $v_j$ . We call a solution  $C$  strongly connected if for every pair  $(v_i, v_j) \in V \times V$  there is a path from  $v_i$  to  $v_j$  and a path from  $v_j$  to  $v_i$  in  $G(C)$ . In this paper, we explore symmetric solutions to the problem, i.e., there is a direct communication from  $v_i$  to  $v_j$  iff there is a direct communication from  $v_j$  to  $v_i$ .

Given a set of circles  $C$ , we denote by

$$A(C) = \bigcup_{i=1}^n A(C_i)$$

the area covered by the circles in  $C$ . The goal is to minimize the size of the area

$$|A(C)| = Vol(A(C))$$

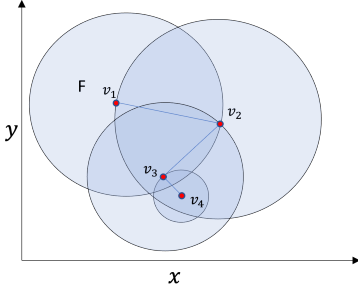
under the condition that the graph  $G(C)$  is strongly connected.

We are considering the problem with a given set  $P_A$  of adversarial nodes which should not be able to listen to the communication, i.e., they should not be included in any of the circles. So, the goal is to compute a set of circles  $C$  such that  $G(C)$  is strongly connected, hence no circles intersects with any adversary in  $P_A$ .

## 3 REPRESENTATION OF SOLUTIONS

We will use spanning trees as representations of possible solutions for the problem that we aim to solve. We first discuss the approach for the case where no adversary is presented and adapt it to the adversarial model in Section 4.

We encode a solution as a spanning tree  $T$  for the complete undirected graph  $G = (V, E)$  given by the set of points and their Euclidean distances. Note, that we work with  $G$  as being an undirected spanning tree  $T$  therefore consists of a set of undirected edges.



**Figure 1: Spanning tree for graph with 4 nodes and resulting 4 circles.**

**DEFINITION 1.** A connected acyclic subgraph  $T = (V, E(T))$  of  $G$  with  $E(T) \subseteq E$  is called a spanning tree.

We denote by  $\mathcal{T}(G)$  the set of all spanning trees of a given graph  $G$ . When the graph  $G$  is clear from the context, we also use  $\mathcal{T}$  instead of  $\mathcal{T}(G)$ .

As we consider complete graphs in the plane the total number of spanning trees of  $G$  is  $n^{n-2}$ . The minimum spanning tree problem is a well-known combinatorial optimization problem on graphs and is used in various applications [10, 14]. Computing a minimum spanning tree for a given graph can be done efficiently by using Kruskal's algorithm [16] in time  $O(n^2 \log n)$ .

**DEFINITION 2.** A spanning tree  $T^*$  is called a minimum spanning tree (MST) if its sum of distances is minimal across the set of all possible spanning trees  $\mathcal{T}$ , i.e.

$$T^* = \arg \min_{T \in \mathcal{T}} \sum_{e \in E(T)} d(e).$$

We set the radius of  $i$  for the solution  $T$  to

$$r_i(T) = \max_{j \in N_T(v_i)} d(v_i, v_j)$$

where  $N_T(v_i)$  denotes the set of direct neighbours of  $v_i$  in  $T$ . For an illustration see Figure 1.

Given a solution  $T$  represented as a spanning tree with the defined values of  $r_i$ , the solution is obviously strongly connected, i.e., communication is possible between any pair of nodes in both directions.

Let  $A(v_i, r_i(T))$  is the area covered by point  $v_i$  with radius  $r_i(T)$ . We denote by

$$A(T) = \bigcup_{i=1}^n A(v_i, r_i(T))$$

the area covered by spanning tree  $T$ .

The goal is to minimize the size of the area

$$|A(T)| = \text{Vol}(A(T)),$$

where  $\text{Vol}(A)$  denotes the volume of a given area  $A$ , under the condition that  $T$  is strongly connected.

**DEFINITION 3 (MINIMUM AREA SPANNING TREE (MAST) PROBLEM).** An area minimum spanning tree  $T^*$  is a spanning tree whose covered area across the set of all possible spanning trees denoted as  $\mathcal{T}$  is minimal, i.e.  $T^* = \arg \min_{T \in \mathcal{T}} |A(T)|$ .

## 4 HANDLING ADVERSARIES

We now show how to adapt the minimum area spanning tree problem to the case where adversaries are present. Assume the position of the given receiver from the adversary to detect the footprint is exposed in the map. Hereby, while we construct the communication area by MAST, we need to avoid the exposed receiver (i.e., Figure 2).

With the constraints of additional points  $P_A$ , the advanced scenario is defined as an adversary model for the detection and concealment of spatially defined communication networks. We mathematically generalised such scenario. Let  $G = (V, E)$  be a directed connected graph with node  $V$ , edge set  $E$  and Euclidean distance  $d : E \rightarrow R_+$  which assigns positive distance between each edge. Given a set of points  $P_A$ , the goal of the problem is to find a minimum area spanning forest that covers all the nodes from  $V$  and avoids covering the points from  $P_A$ .

Note that we are referring to a spanning forest as  $G$ , which may not have a connected subgraph that avoids all adversaries in  $P_A$ . Formally, the problem is given in the following definition.

**DEFINITION 4 (ADVERSARIAL MINIMUM AREA SPANNING FOREST (AMASF) PROBLEM).** An area minimum spanning forest  $F^*$  of the graph  $G = (V, E)$  in the Euclidean plane for a set of adversarial points  $P_A$  is the spanning forest with a minimal number of connected components  $k$  whose covered area across the set of all possible spanning forest with  $k$  components denoted as  $\mathcal{F}$  is minimal and whose area does not intersect any points in  $P_A$ , i.e.

$$F^* = \arg \min_{F \in \mathcal{F}} \{A(F) \mid A(F) \cap P_A = \emptyset\}.$$

Given a set  $P_A$  of adversarial nodes, we obtain the undirected graph  $G_A = (V, E_A)$  from  $G$  by deleting all edges impacted by an adversary. Formally we have

$$E_A = \{e = \{v_i, v_j\} \in E \mid \nexists p_k \in P_A : d(v_i, p_k) < d(v_i, v_j) \wedge d(v_j, p_k) < d(v_j, v_i)\}.$$

We use the graph  $G_A$  for diversity optimization. Let  $k$  be the number of connected components of  $G_A$ . Then Kruskal's algorithm returns a minimal spanning forest  $MSF$  of  $G_A$ . We use  $\mu$  copies of  $MSF$  for the initial population.

## 5 EDO FOR AMASF

We consider the  $(\mu + \lambda)$ -EA shown in Algorithm 1 which has already been used in the context of evolutionary diversity optimization [1, 23]. We adapt this algorithm to the AMASF problem by using our own representation, mutation, and fitness functions to tackle the problem.

We use the graph  $G_A$  for diversity optimization. Let  $k$  be the number of connected components of  $G_A$ . Then Kruskal's algorithm returns a minimal spanning forest  $MSF$  of  $G_A$ . Let  $A(MSF)$  be the size of the area of the minimum spanning forest. As quality criteria for diversity optimization, we consider solutions with the same number of connected components as  $F$  having area at most  $(1 + \alpha) \cdot |A(MSF)|$ .

The  $(\mu + \lambda)$ -EA maintains at each generation a population  $P$  of  $\mu$  spanning forests which all meet a given quality criterion, i.e.,  $A(F) \leq (1 + \alpha) \cdot |A(MSF)|$  holds for all individuals  $F \in P$ . The algorithm starts with a population of  $\mu$  spanning forest that meet the quality criterion. For our experiments, we obtain this population

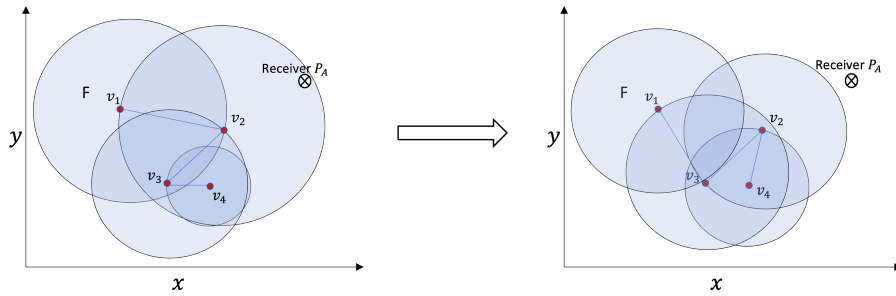


Figure 2:  $P_A$  presented the detection receiver. To avoid it, the edge between  $v_1$  and  $v_2$  should be removed.

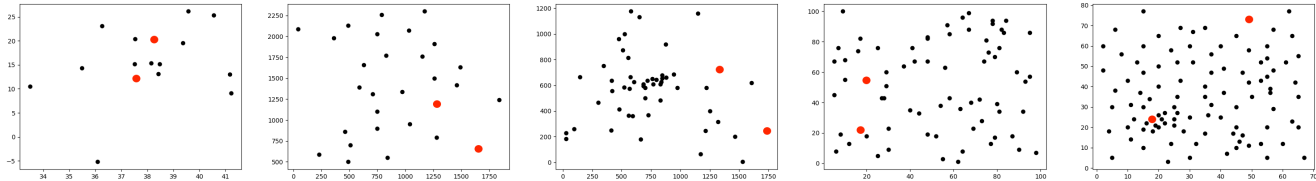


Figure 3: Illustration of instances *ulysses16*, *bays29*, *berlin52*, *st70* and *eil101* and adversary points  $P_A = 2$  shown in red, respectively.

by computing  $\mu$  copies of a minimum spanning forest. An offspring population  $C$  of size  $\lambda$  is created by mutating  $\lambda$  individuals of  $P$ . A newly created individual  $F'$  is added to the population  $P$  if it meets the area quality criterion. In the survival selection step, the algorithm always keeps a feasible individual  $F^*$  which has the smallest covered area and removes individuals with the smallest contribution to diversity greedily until  $P$  is again of size  $\mu$ . As a baseline, we investigate the case  $\lambda = 1$  which results in the  $(\mu + 1)$ -EA for diversity optimization. As our population is focused on optimizing diversity, this often already leads to good results. For comparison, we also present results for larger offspring population size.

### 5.1 Mutation operators

We are working with spanning forest as the representation of solutions and consider different mutation operators applied to the spanning trees of the given forest. Different mutation operators for spanning trees have been considered in the literature. They all make use of the fact that a new spanning tree can be obtained from a given spanning tree by introducing an edge currently not part of the solution and removing an edge from the resulting cycle. For the removal of an edge from a given cycle we are using the biased diversity edge removal operator given in Algorithm 2 which gives preference to the removal of edges appearing more frequent in the population.

Our mutation operators differ in the way that they choose edges to be inserted. We consider the standard uniform edge mutation operator shown in Algorithm 3 as well as a biased edge mutation operator giving preferences to edges of smaller costs (see Algorithm 4) as done in [27]. Furthermore, we consider a biased edge mutation operator that gives preferences to edges currently less present in the population (see Algorithm 5) and therefore trying to focus on increasing diversity. As the measure for diversity for an edge  $e$ , we use  $d(e) = 1/(c(e) + 1)$ , where  $c(e)$  is the number

---

### Algorithm 1 ( $\mu + \lambda$ ) EA for Diversity Optimization

---

- 1: Initialize the population  $P$  with  $\mu$  spanning forests such that  $|A(F)| \leq (1 + \alpha) \cdot |A(MSF)|$  for all  $F \in P$ .
  - 2: **while** termination criterion is not reached **do**
  - 3:   let  $C \subseteq P$  where  $|C| = \lambda$ .
  - 4:   **for**  $F \in C$  **do**
  - 5:     produce an offspring  $F'$  of  $F$  by applying mutation.
  - 6:     **if**  $|A(F')| \leq (1 + \alpha) \cdot |A(MSF)|$  **then**
  - 7:       add  $F'$  to  $P$ .
  - 8:     **end if**
  - 9:   **end for**
  - 10:    $F^* = \arg \min_{F \in P} |A(F)|$
  - 11:   **while**  $|P| > \mu$  **do**
  - 12:     let  $\hat{F} = \arg \max_{F \in P \setminus \{F^*\}} D(P \setminus \{F\})$ .
  - 13:     remove  $\hat{F}$  from  $P$ .
  - 14:   **end while**
  - 15: **end while**
  - 16: return  $P$ .
- 

of occurrences of edge  $e$  in the population  $P$ . Note that the value of  $d(e)$  increases with decreasing value of  $c(e)$ . Each operator is carrying out sequentially  $k$  times to produce an offspring  $F'$  from  $F$  where  $k$  is chosen according to  $1 + Pois(1)$  where  $Pois(1)$  denotes the Poisson distribution with expected value 1. This gives the mutation the possibility of carrying out more than one basic operation.

### 5.2 Area Calculation

Given a forest  $F$  representing a possible solution to the AMASF problem, we need to calculate the area  $|A(F)|$  of the circles determined by  $F$ . Previous approaches used Monte Carlo sampling [4, 8] which is only approximate and computationally expensive. In the following, we show how to compute the area exactly and more efficiently. To



**Algorithm 2** Biased Diversity Edge Removal

- 1: Let  $\hat{C}$  be a given cycle in a solution  $F$  and  $c(e)$  be the number of occurrences of edge  $e$  in  $P$ ,
- 2: Select an edge  $e \in \hat{C}$  with probability  $\frac{c(e)+1}{\sum_{e \in \hat{C}} c(e)+1}$  and remove  $e$  from  $F$ .

**Algorithm 3** Uniform Edge Mutation Operator

- 1: Obtain  $F'$  from  $F$  by introducing an edge  $e \in E_A \setminus F$  chosen uniformly at random into  $F$  and removing an edge  $e'$  using Algorithm 2 from the resulting cycle.

**Algorithm 4** Biased Quality Mutation Operator

- 1: Let  $\hat{E} = E_A \setminus F$ . Assign ranks to the edges  $e = (v_i, v_j) \in \hat{E}$  in increasing order of their distances  $d(v_i, v_j)$  and let  $a = \frac{|\hat{E}|}{|\hat{E}|+1}$ .
- 2: Select an edge  $e^r \in \hat{E}$  of rank  $r$  with probability  $a^r / \left( \sum_{r=1}^{|\hat{E}|} a^r \right)$ .
- 3: Obtain  $F'$  from  $F$  by introducing  $e^r$  into  $F$  and removing an edge  $e'$  using Algorithm 2 from the resulting cycle.

**Algorithm 5** Biased Diversity Mutation Operator

- 1: Let  $\hat{E} = E_A \setminus F$ ,  $c(e)$  be the number of occurrences of edge  $e$  in  $P$ , and  $d(e) = 1/(1+c(e))$  for all  $e \in \hat{E}$ .
- 2: Select an edge  $e \in \hat{E}$  with probability  $\frac{d(e)}{\sum_{r=1}^{|\hat{E}|} d(e)}$ .
- 3: Obtain  $F'$  from  $F$  by introducing  $e$  into  $F$  and removing an edge  $e'$  using Algorithm 2 from the resulting cycle.

calculate the area of union circles, we use Green's Theorem [12] which gives an exact area calculation in time  $O(n^2 \log(n))$ .

Given a 2D vector field  $F$ , a region  $R$  in the  $x-y$  plane and the anticlockwise boundary  $C$  of the region  $R$ . Green's theorem [12] states that the *double line integral of the curl of  $F$  within  $R$  equals the line integral of  $F$  around the boundary of  $R$*  as defined in Equation 1:

$$\int \int_R 2d - \text{curl } \mathbf{F} \, dA = \oint_C \mathbf{F} \cdot d\mathbf{r} \quad (1)$$

Alternatively,  $\mathbf{F}$  can be written (component-wise) as  $\mathbf{F}(x, y) = P(x, y)\hat{i} + Q(x, y)\hat{j}$ , thus the theorem can be written as

$$\int \int_R \left( \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dA = \oint_C P \, dx + Q \, dy \quad (2)$$

If  $\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}$  is set to 1 then the area of the region  $R$  equals the right hand side of Equation 2 [26]. To find the union area of a set of union circles, we first remove all circles that are fully inside another circle to avoid duplicate areas. We integrate along the outer border of the union circles in an anticlockwise direction, forming the area of the region to the left of the path, which is the entire area contained inside the path. To remove the area that should not be included in the final area, we integrate along the closed path in a clockwise direction, resulting in the desired area of the union of the given circles.

### 5.3 Diversity Measure

We utilise a diversity measure that is based on the edge overlap of spanning trees which has been used in a study on evolutionary diversity optimization for the minimum spanning tree problem [3] and adapt it to spanning forests with  $k$  connected components. The diversity measure works as follows. For two spanning forests  $F_i$  and  $F_j$ , we define the edge overlap as the number of edges that are shared, i.e.,

$$o(F_i, F_j) := |F_i \cap F_j|.$$

For a population  $P = \{F_1, \dots, F_\mu\}$  of  $\mu$  spanning forests, we minimize the sum of the pairwise edge overlaps in the different solutions of  $P$  by maximizing the diversity score of  $P$  given as

$$D(P) := \mu(\mu-1)(n-k) - \sum_{i=1}^{\mu} \sum_{\substack{j=1 \\ j \neq i}}^{\mu} o(F_i, F_j) \quad (3)$$

where  $k$  is the number of connected components of the graph.

## 6 EXPERIMENTAL INVESTIGATIONS

In this section, we investigate the  $(\mu+1)$ -EA with different mutations on several benchmarks and compare them to the  $(\mu+\lambda)$ -EA algorithm with uniform edge mutation, and the new bias quality and bias diversity mutation operators. Additionally, we investigate the impact of different  $\alpha$  values and different parent population sizes.

### 6.1 Experimental Settings

To the best of our knowledge, there are no benchmark instances that include adversary points available in the literature. To carry out our investigations, we consider popular instances of the traveling salesperson problem (TSP) in the Euclidean plane from TSPLIB<sup>1</sup>. We obtain from them complete graphs  $G = (V, E)$  of small size with  $n \in \{16, 29, 52, 70\}$  and graphs of medium size with  $n \in \{101, 124, 150, 200\}$ . Therefore, our approaches were tested with TSPLIB instances and real-world networks DIMACS10<sup>2</sup>, Gnutella (09)<sup>3</sup>, constructed subgraph  $n \in \{105, 115\}$ , respectively. We chose a random subset  $P_A$  of the points from the given TSP instances and used them as adversaries instead of standard input points. Figure 3 visualises examples of the used instances in Euclidean space obtained from the TSP instances ulysses16, bay29, st70, eil101 by choosing two adversarial points.

For our experiments, we carry out 30 runs for each approach on each instance. Each run consists of 10 000 fitness evaluations. We compute the final population diversity and the best area obtained by each considered algorithms, and report the average, maximum, minimum results, and standard deviation for these 30 runs.

We study the performance of  $(\mu+\lambda)$ -EA with values of  $\mu = 5, 20, \lambda = 1, 2, 5, 10$ , and uniform edge, bias quality and bias diversity mutation. We consider  $\alpha = 0.05, 0.1, 0.2, 0.3, 0.4$  for our experimental investigations. An important characteristic of the final solutions is that the solutions have to fulfill a minimum-quality criterion determined by the parameter  $\alpha$ .

<sup>1</sup><http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>

<sup>2</sup><https://www.cc.gatech.edu/dimacs10/archive/clustering.shtml>

<sup>3</sup><http://snap.stanford.edu/data/p2p-Gnutella09.html>

**Table 1: Maximum (max), minimum (min), mean (mean), and standard deviation (std) in terms of area calculation size for (5 + 1)-EA and (5 + 2)-EA with uniform edge mutation operator and  $\alpha = 0.2$ .**

	(5 + 1)-EA (1)				(5 + 2)-EA (2)			
	max	min	mean AREA	std	max	min	mean AREA	std
ulysses16	32.71426114	32.71426114	32.71426114	1.44624E-14	32.71426114	32.71426114	32.71426114	1.44624E-14
bays29	3155645.527	3084725.959	3104257.276	17267.31246	3187884.8766	3090029.3624	3170603.518	32953.40117
berlin52	2059638.890	2059638.89	2059638.8900	1.42171E-09	2059652.4551	2059652.4551	2059652.455	1.42171E-09
st70	9783.037891	9653.864724	9746.245919	37.60046579	9783.037891	9599.700145	9774.389587	35.79726272
eil101	6431.819816	6383.858888	6413.699712	13.97422923	6432.348379	6432.348380	6432.348379	1.85119E-12
DIMACS10	1064089.550	1060994.700	1062594.110	1246.303900	1064089.550	1064089.550	1064089.550	2.3681E-10
Gnutella115	132799.2080	126737.3830	129097.2180	1430.684620	133165.8860	128259.3340	132832.9970	1196.938180
kr124	84337128.88	82481464.68	83835776.60	547406.4062	84337128.88	84190842.44	84329447.12	30200.72499
ch150	473311.7123	472171.0425	473249.0724	230.1580423	2473311.7123	473311.7123	473311.7123	2.9619E-10
kroA200	7743569.727	7706700.591	7740564.446	8888.726571	7743569.727	7743569.727	7743569.727	1.72942E-09

**Table 2: Maximum (max), minimum (min), mean standard deviation (std) in terms of diversity value for (5 + 1)-EA (1) and (5 + 2)-EA (2) with uniform edge mutation and  $\alpha = 0.2$ .**

	(5 + 1)-EA (1)				(5 + 2)-EA (2)			
	max	min	mean DIV	std	max	min	mean DIV	std
ulysses16	16	16	16	0	16	12	15.6	1.220514306
bays29	360	348	353.0666667	2.958719823	364	342	357.1724138	3.274020743
berlin52	828	794	813.7333333	6.862860233	836	802	816.2000000	7.378440168
st70	932	890	911.2000000	11.18619839	968	930	944.0666667	9.475679698
eil101	1674	1642	1656.4000000	8.143539856	1656	1632	1645.2666667	6.549194914
DIMACS10	1660	1506	1582.0000000	36.68484400	1642	1470	1579.2666700	37.67184380
Gnutella115	2004	1956	1988.1333000	11.964699000	1996	1966	1983.9310300	7.699401810
kr124	1896	1794	1848.689655	24.30329703	1912	1812	1859.517241	23.64411840
ch150	2190	1916	2073.862069	65.75915153	2178	2100	2114.965517	46.47463422
kroA200	2584	2208	2406.413793	93.70070790	2660	2222	2465.724138	121.0769816

**Table 3: Maximum (max), minimum (min), mean standard deviation (std) in terms of area calculation value for (20 + 5)-EA (1) and (20 + 10)-EA (2) with uniform edge mutation and  $\alpha = 0.2$ .**

	(20 + 5)-EA (1)				(20 + 10)-EA (2)			
	max	min	mean AREA	std	max	min	mean AREA	std
ulysses16	32.71426114	32.71426114	32.71426114	1.44624E-14	32.71426114	32.71426114	32.71426114	1.44624E-14
bays29	3187884.877	3086165.663	3123922.316	27193.79872	3187884.877	3110893.694	3173803.203	21925.31717
berlin52	2059652.455	2059638.89	2059650.584	4.760381771	2059652.455	2059638.89	2059650.584	4.760381771
st70	9783.037891	9771.741638	9782.590533	2.109623675	9783.037891	9783.037891	9783.037891	7.40475E-12
eil101	6432.34838	6432.298621	6432.346664	0.009240014	6432.34838	6431.972343	6432.335413	0.069828318

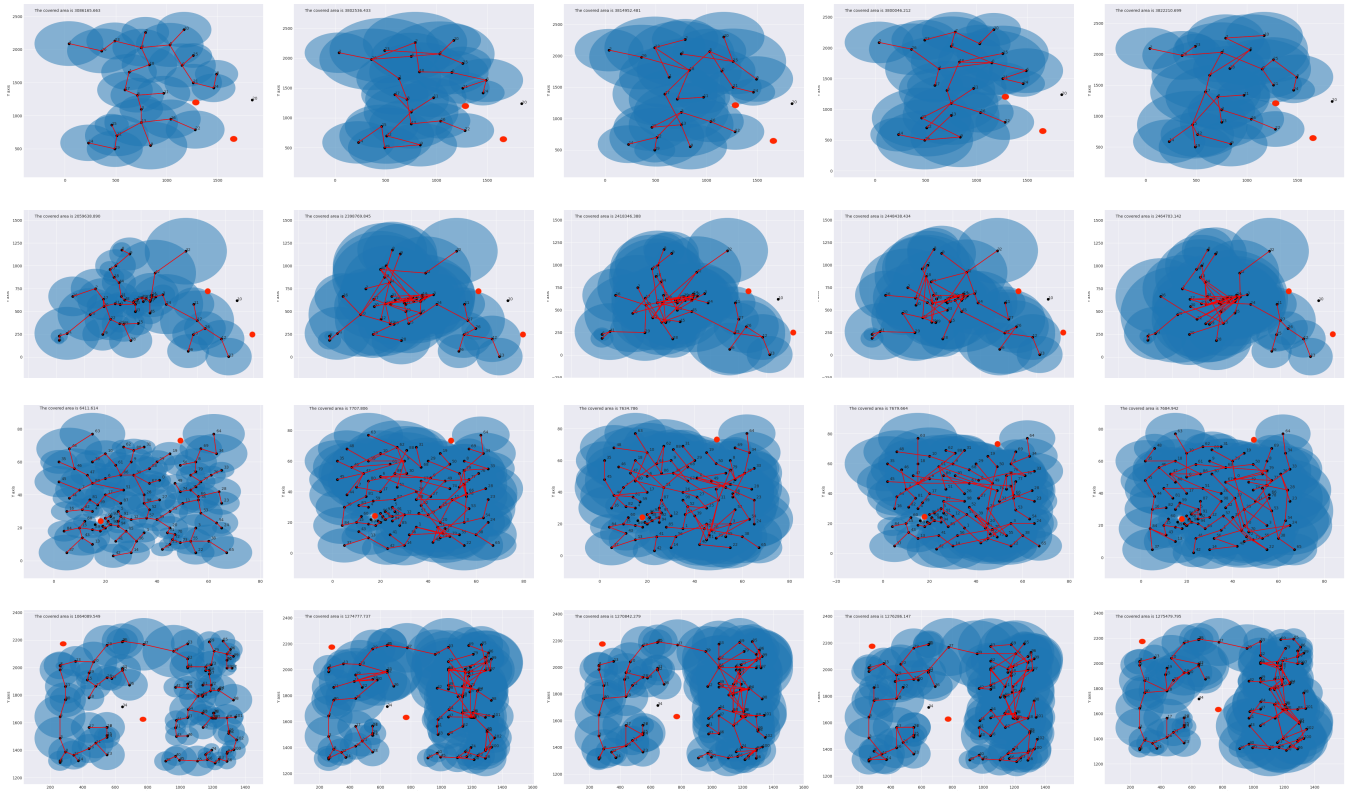
**Table 4: Maximum (max), minimum (min), mean standard deviation (std) in terms of diversity value for (20 + 5)-EA (1) and (20 + 10)-EA (2) with uniform edge mutation and  $\alpha = 0.2$ .**

	(20 + 5)-EA (1)				(20 + 10)-EA (2)			
	max	min	mean DIV	std	max	min	mean DIV	std
ulysses16	266	262	266	0.73029674	266	234	266.6	8.173126697
bays29	6038	5874	6000.866667	34.75900611	6080	5900	6018.275862	36.90808714
berlin52	14820	14324	14571.51724	122.8770061	14750	14276	14542.96552	119.5737915
st70	15486	14896	15288.68966	129.7616231	15744	14966	15373.51724	177.1816301
eil101	28660	28330	28518.48276	89.04357709	28682	28306	28497.10345	106.0923940

## 6.2 Main Results

We first consider results for the (5+1)-EA and the (5 + 2)-EA for area optimization and diversity optimization using uniform edge mutation for  $\alpha = 0.2$ . Note that the (5 + 2)-EA is only allowed to carry out 5,000 generations as it produces 2 offspring in each generation. Our goal is to see how this slight increase in offspring

population size affects the results of our EDP approach. The experimental results are shown in Table 1 and 2. The results show that the (5 + 1)-EA is able to achieve lower average mean values for the all instances in terms of the obtained area in all cases. It can be observed that the ( $\mu + 2$ )-EA obtains the lowest mean values for the instance with 16 nodes. In contrast, the (5 + 2)-EA obtains



**Figure 4: Visualised final populations for instances bays29, berlin52, eil101 and real-word network DIMACS10 for  $\mu = 5$ ,  $\lambda = 1$ ,  $\alpha = 0.2$ , and two adversaries shown in red.**

the highest mean value in terms of diversity in most of the cases for the instance with 16, 29, 70, 124, 150, 200 nodes (see Table 2). This shows that the amount of diversity can be increased by even slightly increasing the offspring population size from 1 to 2. We now consider the optimization results for diversity and area obtained by  $(5 + 1)$ -EA using uniform edge mutation for different values of  $\alpha$ . Our goal is to see how  $\alpha$  affects the diversity score that can be obtained by the algorithm. Table 5 shows a comparison in terms of diversity  $DIV$  and area calculation  $AREA$  for  $\alpha = 0.05, 0.1, 0.3, 0.4$  for all benchmark instances. Our results show that the amount of diversity that can be obtained in the population is dependent on the quality threshold  $\alpha$  that is incorporated in the EDO approach. It can be observed that the diversity score increases accordingly with an increase of the  $\alpha$  values. A similar picture can be seen when we compare different  $\alpha$  values in terms of the obtained area values in most of the cases. With increasing  $\alpha$  values, the algorithm is able to achieve smaller area calculation values in some cases, which might be due to a higher diversity in the population, although the differences are relatively small.

### 6.3 Visualization

We now show some final results obtained by the  $(5 + 1)$ -EA using uniform edge mutation and  $\alpha = 0.2$ . Figure 4 shows an example construction of our final populations that occurred by investigating our EDO approach for the instances bays29, berlin52, eil101 and DIMACS10. The red points represent the randomly located two adversarial points. A minimum spanning tree is shown by the red

lines. We can observe five different spanning trees constellations for each instance that visualise the covered area by the senders under the condition that communication across the network is guaranteed despite the presence of two adversaries. We observe that all different scenarios are able to maintain a covered area while capturing the most possible diverse solutions.

### 6.4 Impact of Mutation Operators

In addition, we compare the effect of using the new bias quality mutation (BQ) approach outlined in Algorithm 4 and bias diversity mutation (BD) approach outlined in Algorithm 5 within the  $(5 + 1)$ -EA. The results for the improved EDO approaches are shown in Table 6. It can be observed that the bias quality mutation can improve the diversity result and area calculation for  $(5 + 1)$ -EA for all of the investigated settings compared to the bias diversity mutation. This allows us to draw the conclusion that there is an advantage to giving preferences to edges of smaller costs during the mutation step.

In contrast, diversity score results are lower when using the bias diversity mutation for smaller graphs than with the uniform edge mutation. Comparing the results, it can be observed that the bias quality mutation shows better results for some cases in terms of diversity and area compared to the uniform edge mutation. Overall, this shows that the mutation operators that are specifically tailored to our specific problem can help to achieve better results although they are not beneficial in all cases.



**Table 5: Maximum (max), minimum (min), mean (mean), and standard deviation (std) in terms of diversity value for (5 + 1)-EA with different  $\alpha$  values, i.e.,  $\alpha = 0.05, 0.1, 0.3, 0.4$  with uniform edge mutation operator.**

	$\alpha$	max	min	mean DIV	std	max	min	mean AREA	std
ulysses16	0.05	16	16	16	0	32.71426114	32.71426114	32.71426114	1.44538E-14
bays29	0.05	274	260	268.4	3.944441746	3129995.064	3084725.959	3103702.06	14212.4975
berlin52	0.05	746	714	732.2	8.24370188	2059638.890	2059638.890	2059638.890	4.73622E-10
st70	0.05	674	546	612.4666667	28.99076349	9783.037891	9633.665855	9749.658692	38.5757768
eil101	0.05	1478	1406	1450.933333	16.27973848	6430.331415	6381.401944	6413.097868	15.56829924
ulysses16	0.1	16	16	16	0	32.71426114	32.71426114	32.71426114	1.44538E-14
bays29	0.1	318	304	309.8666667	3.598211561	3144558.341	3084725.959	3106188.02	15487.71252
berlin52	0.1	788	760	771.8	6.310363671	2059638.89	2059638.89	2059638.89	4.73622E-10
st70	0.1	794	718	761.8	18.34797743	9783.037891	9611.442579	9749.825501	46.14789184
eil101	0.1	1582	1540	1561.133333	12.08799157	6428.989212	6374.377222	6414.156749	13.10189365
ulysses16	0.3	16	16	16	0	32.71426114	32.71426114	32.71426114	1.44538E-14
bays29	0.3	380	374	376.3448276	1.609699663	3123360.178	3084725.959	3104282.627	12218.57865
berlin52	0.3	846	814	832.4827586	7.55277002	2059638.890	2059638.890	2059638.890	4.73904E-10
st70	0.3	996	956	982.4	8.794982896	9783.037891	9633.665855	9747.346167	46.19794731
eil101	0.3	1692	1676	1683.241379	4.014751126	6430.328751	6383.123056	6409.412473	14.63409373
ulysses16	0.4	16	16	16	0	32.71426114	32.71426114	32.71426114	1.44624E-14
bays29	0.4	390	386	387.7333333	1.25762045	3141022.805	3084725.959	3100377.519	14703.09581
berlin52	0.4	858	830	846.9655172	4.586336529	2059638.890	2059638.890	2059638.890	4.73904E-10
st70	0.4	1030	1008	1020.344828	5.633293011	9783.037891	9547.218612	9734.657696	56.88975241
eil101	0.4	1702	1680	1694.137931	4.98371733	6430.200925	6381.339747	6412.21713	13.47417665

**Table 6: Maximum (max), minimum (min), mean (mean), and standard deviation (std) in terms of diversity value and area calculation size for (5 + 1)-EA with  $\alpha = 0.2$ .**

	mutation	max	min	mean DIV	std	max	min	mean AREA	std
ulysses16	BQ	16	16	16	0	32.71426114	32.71426114	32.71426114	1.44624E-14
bays29	BQ	360	346	354.8965517	3.569161842	3110893.694	3084725.959	3088472.429	8200.898583
berlin52	BQ	800	774	783.862069	9.530867664	2059638.890	2059638.890	2059638.890	4.73904E-10
st70	BQ	954	912	937.2413793	10.24574885	9727.764339	9488.765958	9636.056363	70.95203904
eil101	BQ	1794	1744	1775.724138	10.81961364	6428.238216	6371.336745	6412.874553	13.37907717
ulysses16	BD	16	8	11.6	2.943139305	32.71426114	32.71426114	32.71426114	1.44624E-14
bays29	BD	358	346	353.8571429	2.490192402	3164049.389	3084725.959	3124600.814	20957.97462
berlin52	BD	834	796	817.2413793	8.592584728	2059639.641	2059638.890	2059638.928	0.152408731
st70	BD	892	802	859.2413793	20.39197176	9783.037891	9686.119679	9766.99635	26.69093431
eil101	BD	1786	1725	1759.448276	15.39848371	6432.34838	6416.992013	6430.301149	3.998170465

## 6.5 Larger Population Sizes

Furthermore, we provide results for larger parent and offspring population sizes using in the  $(\mu + \lambda)$ -EA with uniform edge mutation, i.e.,  $\mu = 20, \lambda = 5, 10$ . Again we provide results for  $\alpha = 0.2$ . Table 3 shows the results in terms of the size of the area that is obtained. The results show that the  $(20 + 10)$ -EA obtains the lowest mean values compared to the results obtained by the  $(20 + 5)$ -EA in terms of area in most of the cases. This indicates that the parent population sizes play an important role in terms of diversity optimization for minimizing the size of the area under the condition that the graphs are strongly connected. Table 4 shows a comparison for the  $(20 + \lambda)$ -EA in terms of population diversity *DIV* for  $\lambda = 5$  and  $\lambda = 10$ . The  $(20 + 10)$ -EA obtains the highest mean value in terms of diversity for the instances bays29 and st70.

## 7 CONCLUSIONS

Computing high quality and a diverse set of solutions provides decision makers with different scenarios with high quality solutions that all meet a given quality criterion. In recent years, this area has gained increasing interest in the evolutionary computation literature. We considered the concealment of spatially defined communication networks with a given set of adversaries. Previous brute force methods were only able to tackle small instances by

solving the MAST problem and did not take into account the presence of adversaries. We formulated the problem in the presence of adversaries as the adversarial minimum area spanning forest (AMASF) problem and designed EDO approaches for the problem that compute diverse sets of solution for a given quality threshold parameter. Our results show that evolutionary diversity optimization can be used to effectively provide diverse sets of high quality solutions for the concealment of communication networks in large settings. This work also sets the basis for generating pre-prepared robust options in a dynamic adversary case which we consider as an important topic for future work.

## ACKNOWLEDGMENTS

This work has been supported by the Australian Research Council (ARC) through grants DP190103894 and FT200100536. We acknowledge funding from Defence Science and Technology's AI4DM program.

## REFERENCES

- [1] Bradley Alexander, James Kortman, and Aneta Neumann. 2017. Evolution of artistic image variants through feature based diversity optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017*. ACM, 171–178. <https://doi.org/10.1145/3071178.3071342>
- [2] Thomas Bäck, Michael Emmerich, and Ofer M. Shir. 2008. Evolutionary algorithms for real world applications [Application Notes]. *IEEE Comput. Intell. Mag.* 3, 1 (2008), 64–67. <https://doi.org/10.1109/MCI.2007.913378>

- [3] Jakob Bossek and Frank Neumann. 2021. Evolutionary diversity optimization and the minimum spanning tree problem. In *GECCO '21: Genetic and Evolutionary Computation Conference, Lille, France, July 10-14, 2021*. ACM, 198–206. <https://doi.org/10.1145/3449639.3459363>
- [4] Benjamin Campbell, Anthony Perry, Robert Hunjet, Guangsong Wang, and Bruce Northcote. 2018. Minimising RF detectability for low probability of detection communication. In *2018 Military Communications and Information Systems Conference (MilCIS)*. 1–6. <https://doi.org/10.1109/MilCIS.2018.8574116>
- [5] Paz Carmi, Matthew J. Katz, and Joseph S. B. Mitchell. 2006. The minimum-area spanning tree problem. *Comput. Geom.* 35, 3 (2006), 218–225. <https://doi.org/10.1016/j.comgeo.2006.03.001>
- [6] Raymond Chiong, Thomas Weise, and Zbigniew Michalewicz (Eds.). 2012. *Variants of evolutionary algorithms for real-world applications*. Springer. <https://doi.org/10.1007/978-3-642-23424-8>
- [7] David W. Corne, Kalyanmoy Deb, Joshua D. Knowles, and Xin Yao. 2012. Selected aspects of natural computing. In *Handbook of Natural Computing*. Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok (Eds.). Springer, 1737–1801. [https://doi.org/10.1007/978-3-540-92910-9\\_52](https://doi.org/10.1007/978-3-540-92910-9_52)
- [8] Andrew Coyle, Benjamin Campbell, and Robert Hunjet. 2020. Minimising the network detection probability using autonomous vehicles. In *2020 Military Communications and Information Systems Conference (MilCIS)*. IEEE, 1–6.
- [9] Kalyanmoy Deb. 2012. *Optimization for engineering design - algorithms and examples, Second Edition*. PHI Learning Private Limited.
- [10] Narsingh Deo. 2017. *Graph theory with applications to engineering and computer science*. Courier Dover Publications.
- [11] Anh Viet Do, Mingyu Guo, Aneta Neumann, and Frank Neumann. 2022. Analysis of Evolutionary Diversity Optimization for Permutation Problems. *ACM Trans. Evol. Learn. Optim.* 2, 3 (2022), 11:1–11:27. <https://doi.org/10.1145/3561974>
- [12] Herbert Federer. 1945. The Gauss-Green theorem. *Trans. Amer. Math. Soc.* 58, 1 (1945), 44–76.
- [13] Wanru Gao, Samadhi Nallaperuma, and Frank Neumann. 2016. Feature-based diversity optimization for problem instance classification. In *Parallel Problem Solving from Nature - PPSN XIV - 14th International Conference, 2016, Proceedings (Lecture Notes in Computer Science, Vol. 9921)*. Springer, 869–879. [https://doi.org/10.1007/978-3-319-45823-6\\_81](https://doi.org/10.1007/978-3-319-45823-6_81)
- [14] Ronald L. Graham and Pavol Hell. 1985. On the history of the minimum spanning tree problem. *Annals of the History of Computing* 7, 1 (1985), 43–57.
- [15] Mark P. Kleeman, Gary B. Lamont, Kenneth M. Hopkinson, and Scott R. Graham. 2007. Multiobjective evolutionary algorithms for designing capacitated network centric communications. In *Genetic and Evolutionary Computation Conference, GECCO 2007, Proceedings*. ACM, 905. <https://doi.org/10.1145/1276958.1277135>
- [16] Joseph B Kruskal. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society* 7, 1 (1956), 48–50.
- [17] Joel Lehman and Kenneth O. Stanley. 2011. Evolving a diversity of virtual creatures through novelty search and local competition. In *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011*. ACM, 211–218.
- [18] David Lynch, Takfarinas Saber, Stepán Kucera, Holger Claussen, and Michael O'Neill. 2019. Evolutionary learning of link allocation algorithms for 5G heterogeneous wireless communications networks. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*. ACM, 1258–1265. <https://doi.org/10.1145/3321707.3321853>
- [19] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *CoRR abs/1504.04909* (2015). <http://arxiv.org/abs/1504.04909>
- [20] Aneta Neumann, Denis Antipov, and Frank Neumann. 2022. Coevolutionary Pareto diversity optimization. In *GECCO '22: Genetic and Evolutionary Computation Conference, 2022*. ACM, 832–839. <https://doi.org/10.1145/3512290.3528755>
- [21] Aneta Neumann, Jakob Bossek, and Frank Neumann. 2021. Diversifying greedy sampling and evolutionary diversity optimisation for constrained monotone submodular functions. In *GECCO '21: Genetic and Evolutionary Computation Conference, 2021*. ACM, 261–269. <https://doi.org/10.1145/3449639.3459385>
- [22] Aneta Neumann, Wanru Gao, Carola Doerr, Frank Neumann, and Markus Wagner. 2018. Discrepancy-based evolutionary diversity optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*. ACM, 991–998. <https://doi.org/10.1145/3205455.3205532>
- [23] Aneta Neumann, Wanru Gao, Markus Wagner, and Frank Neumann. 2019. Evolutionary diversity optimization using multi-objective indicators. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*. ACM, 837–845. <https://doi.org/10.1145/3321707.3321796>
- [24] Adel Nikfarjam, Aneta Neumann, and Frank Neumann. 2022. Evolutionary diversity optimisation for the traveling thief problem. In *GECCO '22: Genetic and Evolutionary Computation Conference, 2022*. ACM, 749–756. <https://doi.org/10.1145/3512290.3528862>
- [25] Adel Nikfarjam, Aneta Neumann, and Frank Neumann. 2022. On the use of quality diversity algorithms for the traveling thief problem. In *GECCO '22: Genetic and Evolutionary Computation Conference, 2022*. ACM, 260–268. <https://doi.org/10.1145/3512290.3528752>
- [26] Washek F Pfeffer. 1991. The Gauss-Green theorem. *Advances in Mathematics* 87, 1 (1991), 93–147.
- [27] Günther R. Raidl, Gabriele Koller, and Bryant A. Julstrom. 2006. Biased mutation operators for subgraph-selection problems. *IEEE Trans. Evol. Comput.* 10, 2 (2006), 145–156. <https://doi.org/10.1109/TEVC.2006.871251>
- [28] Tamara Ulrich and Lothar Thiele. 2011. Maximizing population diversity in single-objective optimization. In *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011*. ACM, 641–648. <https://doi.org/10.1145/2001576.2001665>
- [29] Shihao Yan, Xiangyun Zhou, Jinsong Hu, and Stephen V Hanly. 2019. Low probability of detection communication: Opportunities and challenges. *IEEE Wireless Communications* 26, 5 (2019), 19–25.