

Evolutionary Diversity Optimization and the Minimum Spanning Tree Problem

Jakob Bossek
Statistics and Optimization
Dept. of Information Systems
University of Münster
Münster, Germany

Frank Neumann
Optimisation and Logistics
School of Computer Science
The University of Adelaide
Adelaide, Australia

ABSTRACT

In the area of evolutionary computation the calculation of diverse sets of high-quality solutions to a given optimization problem has gained momentum in recent years under the term evolutionary diversity optimization. Theoretical insights into the working principles of baseline evolutionary algorithms for diversity optimization are still rare. In this paper we study the well-known Minimum Spanning Tree problem (MST) in the context of diversity optimization where population diversity is measured by the sum of pairwise edge overlaps. Theoretical results provide insights into the fitness landscape of the MST diversity optimization problem pointing out that even for a population of $\mu = 2$ fitness plateaus (of constant length) can be reached, but nevertheless diverse sets can be calculated in polynomial time. We supplement our theoretical results with a series of experiments for the unconstrained and constraint case where all solutions need to fulfill a minimal quality threshold. Our results show that a simple $(\mu + 1)$ -EA can effectively compute a diversified population of spanning trees of high quality.

CCS CONCEPTS

• Theory of computation → Evolutionary algorithms.

KEYWORDS

Evolutionary algorithms, evolutionary diversity optimization, runtime analysis, minimum spanning tree

ACM Reference Format:

Jakob Bossek and Frank Neumann. 2021. Evolutionary Diversity Optimization and the Minimum Spanning Tree Problem. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21)*, July 10–14, 2021, Lille, France. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3449639.3459363>

1 INTRODUCTION

Evolutionary algorithms and other bio-inspired algorithms have successfully been applied to a wide range of challenging design and optimization problems. Evolutionary algorithms use a population of search points in order to solve a given problem. In the area of

single-objective optimization the classical goal is to produce a single solution that maximizes or minimizes a given function f . Diversity plays a crucial role in the design of evolutionary algorithms as it prevents the algorithm from getting stuck in a single local optima. It also enables the use of crossover which assumes that two good and different solutions can be combined to a new even better solution.

Evolutionary diversity optimization aims to produce for a given optimization problem a set of high quality and diverse solutions [20, 21, 32]. A similar approach is taken in the context of quality diversity [19, 25] where high quality designs with different properties are sought. Evolutionary diversity optimization has been applied in the context of creating diverse images with respect to various features [1] as well as to the task of evolving instances for the Traveling Salesperson Problem (TSP) that show performance differences for algorithms solving the TSP [4, 11]. Such instances are important for automated algorithm selection and configuration [16]. Recently, evolutionary diversity optimization has also been applied to create a diverse set of high quality tours for the TSP [8].

In terms of theoretical foundations of evolutionary computing techniques, the area of runtime analysis has played a crucial role over the last 25 years. We refer to [9, 15, 23] for comprehensive presentations on this research area. So far, the effect of populations has been analyzed with respect to benefits or detrimental effects for solving single-objective optimization problems [7, 31]. Furthermore, populations play a crucial role in providing benefits to multi-objective approaches. Recent studies in the context of Pareto optimization have shown that populations are highly beneficial for optimizing monotone submodular functions with different types of constraints [10, 22, 26, 27, 30]. In the context of evolutionary diversity optimization, initial results have been obtained for the classical functions OneMax and LeadingOnes [12]. Furthermore, special instances of the vertex cover has been investigated and runtime results have been achieved [13]. Still, time complexity results in the context of evolutionary diversity optimization are rare. This is due to the fact that the interactions of individuals in a population together with a measure of diversity is very hard to analyse. However, theoretical foundations on well-studied combinatorial optimization problem are of utmost importance in order to get a deeper understanding of the fitness landscapes posed by these kind of problems and implied challenges for randomized search heuristics such as evolutionary algorithms.

1.1 Our contribution

We contribute to the theoretical analysis of evolutionary diversity optimization and contribute to the highly challenging task of understanding the interactions among individuals in a population when

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '21, July 10–14, 2021, Lille, France

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8350-9/21/07...\$15.00

<https://doi.org/10.1145/3449639.3459363>

carrying out evolutionary diversity optimization. In this paper, we consider evolutionary diversity optimization for the minimum spanning tree problem. This classical problem can be solved with polynomial-time algorithms, e.g. Kruskal [17] or Prim [24]. The MST problem has received significant attention in the area of runtime analysis over the years. Furthermore, different variants of the minimum spanning tree problem such as the multi-objective minimum spanning tree problem and degree or diameter constrained versions are \mathcal{NP} -hard.

In the evolutionary diversity optimization process we minimise pairwise edge overlap in the setting where $1 \leq \mu \leq \lfloor \frac{n}{2} \rfloor$ which allows for a decomposition into a set of μ edge-disjoint – and hence maximally diverse – spanning trees. We are interested in obtaining a diverse set of spanning trees in this context and start our investigations for the case where all edges have the same cost. This implies that we are only interested in a diverse population without imposing a quality criterion. Our theoretical investigations point out structural properties when optimising the diversity of spanning trees using evolutionary algorithms and we give a first runtime analysis for the case of a population size of 2 (Theorem 3.2). Afterwards, we carry out experimental investigations for mutation-based evolutionary diversity optimization approaches to see whether simple operators are able to achieve maximal diversity when all edge weights are equal. We show that a $(\mu + 1)$ -EA using single edge exchanges is efficient in finding a μ -size population if no quality requirements are required. The process can be made more efficient if the number of edge exchanges is drawn from a Poisson distribution which enables several edge exchanges in one mutation step. Furthermore, we observe that with $\mu \rightarrow \lfloor \frac{n}{2} \rfloor$ maximum overlap diversity goes hand in hand with reduced diversity with respect to interesting tree properties like the maximum degree or the diameter. Having gained insights into the behavior for the unconstrained case, we turn to the constrained case where we require that each solution in the population can have weight at most $(1 + \alpha) \cdot \text{OPT}$. Here OPT is the weight of a MST and $\alpha > 0$ is a parameter that determines the maximum cost of a spanning tree to be deemed of high quality. We observe that the choice of α is crucial for the amount of diversity that can be achieved as small values of α determine that only a small number of (often very similar) spanning trees meet the quality criterion.

The paper is structured as follows. We introduce the minimum spanning tree problem and the evolutionary diversity optimization approach that we use in this article in Section 2. We investigate some structural properties for diversity optimization of spanning trees and present some runtime results subsequently in Section 3. Afterwards, we carry out experimental investigations in terms of the diversity optimization process for the unweighted case (Section 4) as well as evolutionary diversity optimization for the weighted case under quality restrictions (Section 5). We finish in Section 6 with some concluding remarks and directions for future work.

2 PROBLEM FORMULATION

Let $G = (V, E)$ be a undirected, connected graph with node set V , edge set E and cost function $c : E \rightarrow \mathbb{R}^+$ which assigns positive real-valued costs to each edge. In the following we denote by $n = |V|$, $m = |E|$ the size of the node and edge set respectively.

Algorithm 1: Diversity maximising $(\mu + 1)$ -EA

- 1 Initialise the population P with μ spanning trees such that $c(T) \leq (1 + \alpha) \cdot \text{OPT}$ for all $T \in P$.
 - 2 Choose $T \in P$ uniformly at random and produce an offspring T' of T by applying a single one-edge-exchange mutation.
 - 3 If $c(T') \leq (1 + \alpha) \cdot \text{OPT}$, add T' to P .
 - 4 If $|P| = \mu + 1$, remove exactly one individual T , where $T = \arg \min_{S \in P} D(P \setminus \{S\})$, from P .
 - 5 Repeat steps 2 to 4 until a termination criterion is reached.
 - 6 Return P
-

Furthermore, we frequently use the notation $E(G)$ to refer to the edge set of graph G . We also frequently identify a graph simply by his edge set E for convenience. The Minimum Spanning Tree (MST) problem is a fundamental combinatorial optimization problem on graphs with countless applications [6]. Each connected acyclic subgraph $T = (V, E')$ with $E' \subset E$ is a spanning-tree (ST). A ST T^* is a minimum spanning tree if its sum of weights is minimal across the set of all possible spanning trees \mathcal{T} , i.e.

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

The MST problem is solvable in polynomial time, e.g. by Prim's algorithm [24].

2.1 Evolutionary Diversity Optimization for the MST

In this paper we investigate evolutionary diversity optimization for the MST, i.e. the goal is – given an input graph G – to find a μ -size population P of spanning trees which is maximal with respect to some diversity measure. In the context of *minimum* spanning trees all solutions need to fulfill a minimum-quality criterion. To this end we adopt the notion of approximation quality: a solution T is within the quality threshold if $c(T) \leq (1 + \alpha) \cdot \text{OPT}$ where parameter $\alpha > 0$ steers the quality requirement and OPT is the value of an MST calculated beforehand.

Our study is based on a simple $(\mu + 1)$ -EA which has been studied extensively in the context of diversity optimization [1, 12, 21]. The algorithmic steps are outlined in Algorithm 1. The $(\mu + 1)$ -EA requires a population P of $\mu = |P|$ spanning trees which all meet a given quality criterion, i.e. $c(T) \leq (1 + \alpha) \cdot \text{OPT}$ for some $\alpha > 0$ and all $T \in P$. This may result from the application of a classic MST-algorithm such as Prim [24]. Next, a solution is sampled from P uniformly at random. This solution is subject to a One-Edge-Exchange (1-EX) mutation to produce a single offspring T' . 1-EX has been subject of various studies in the context of (multi-objective) minimum spanning trees [3, 29]. It has the pleasing characteristic of maintaining the spanning tree property. One step includes adding a random edge to the solution at hand; this step closes exactly one cycle. Finally, a random edge is deleted from this cycle to re-establish the spanning tree property. This implies that mutants are guaranteed to be STs again. The algorithm verifies the mutants quality and, if met, adds T' to the population. Eventually a single solution is required to be dropped to maintain a population size of

μ STs. Here, the algorithm removes the individual with the least contribution to the diversity measure. These steps are repeated until a termination condition is met, e.g. a population with known maximum diversity is found (if known) or the function evaluation limit is reached.

2.2 Diversity Measure

Our diversity measure is based on the edge overlap of STs and has been recently used in a study on evolutionary diversity optimization for the Traveling Salesperson Problem [8]. For two trees T_1, T_2 the (*edge*) *overlap* is defined as the number of shared edges, i.e. $o(T_1, T_2) := |E(T_1) \cap E(T_2)|$. In our definition two trees are maximally diverse if they are *edge-disjoint*, i.e. $o(T_1, T_2) = 0$ or equivalently – as diversity is naturally to be maximised – $(n-1) - o(T_1, T_2) = (n-1)$. We generalize this notion to a population $P = \{T_1, \dots, T_\mu\}$ of μ STs by maximising the pairwise overlap

$$D_o(P) := \mu(\mu-1)(n-1) - \sum_{i=1}^{\mu} \sum_{\substack{j=1 \\ j \neq i}}^{\mu} o(T_i, T_j). \quad (1)$$

Note that for fixed n and μ the term $\mu(\mu-1)(n-1)$ is constant and maximising Eq. (1) is equivalent to minimising the pairwise overlap.

We next give a set of theoretical results pointing out that there exist populations which equalize the frequency of the edges. In particular, under the restriction $1 \leq \mu \leq \lfloor \frac{n}{2} \rfloor$, a population with maximum D_o -diversity, termed D_o -maximal in the following, is achievable. Given a population P we denote by $n(e, P) = |\{T \in P \mid e \in T\}|$ the number of STs the edge e is part of. The goal is to equalize $n(e, P)$ for all $e \in E$.

THEOREM 2.1. *For every complete graph $G = (V, E)$ with $n \geq 4$ nodes and every $\mu \geq 1$ there is a population P of μ spanning trees with*

$$\max_{e \in E} n(e, P) - \min_{e \in E} n(e, P) \in \{0, 1\}.$$

PROOF. We provide a constructive proof. The basic idea is to first generate a set H of spanning trees and to fill P with elements of H . We consider two cases: n even and n odd. First let n be even. Then there is a decomposition of G into $h = \frac{n}{2}$ edge-disjoint n -vertex paths [5]. Let H be the set of those paths. Let $\mu = kh + r$ with $r \in [0, h)$. We put each k copies of each $T \in H$ into P . Subsequently, we put an arbitrary subset $H' \subset H$ with $|H'| = r$ into P . Let $E(H')$ be the set of edges used by H' . Then we have $n(e, P) = k + 1$ for all $e \in E(H')$ and $n(e, P) = k$ for all $e \notin E(H')$ yielding the claim. This completes the proof for even n .

Now let n be odd. According to a well-known paper by Alspach et al. [2] there is a decomposition of G into $\lfloor \frac{n}{2} \rfloor = \frac{n-1}{2}$ edge-disjoint Hamiltonian cycles $C = \{C_1, \dots, C_{(n-1)/2}\}$. Note that removing a single edge from a Hamiltonian cycle yields a Hamiltonian path which is a spanning tree. Since the cycles are edge-disjoint, so are the resulting spanning trees. Note further that each node in G has degree 2 in each Hamiltonian cycle C_i . Let $v \in V$ be an arbitrary node and let T be the star graph¹ with center node v ; T is a spanning tree. We denote by e_i^1 and e_i^2 the two adjacent edges of v for every

¹A star graph is a spanning tree with a center node of degree $(n-1)$ and $(n-1)$ leaf nodes.

cycle C_i . We now define a set $H = \{H_1, \dots, H_n\}$ of n spanning trees as follows:

- $H_i = C_i \setminus \{e_i^1\}$ for $i = 1, \dots, \frac{n-1}{2}$
- $H_{(n-1)/2+1} = T$ and
- $H_{(n-1)/2+1+i} = C_i \setminus \{e_i^2\}$ for $i = 1, \dots, \frac{n-1}{2}$.

T is constructed in a way such that it shares each one edge with every spanning tree in $H \setminus \{T\}$. Likewise, the spanning trees H_i and $H_{(n-1)/2+1+i}$, $i = 1, \dots, \frac{n-1}{2}$ share each $n-2$ edges. Note further that $H_1, \dots, H_{(n-1)/2}$ are edge-disjoint and so are $H_{(n-1)/2+2}, \dots, H_n$ (see Figure 1 for an illustration). As a consequence $n(e, H) = 2$ for all $e \in E$. It follows by construction of H that for each $1 \leq \mu \leq n$ the set $P_\mu = \{H_1, \dots, H_\mu\} \subset H$ satisfies

$$\alpha(P_\mu) := \max_{e \in E} n(e, P_\mu) - \min_{e \in E} n(e, P_\mu) \in \{0, 1\}.$$

with $\alpha(P_n) = 0$.

Now consider $\mu \geq 1$. We construct a μ -size population P by setting $P_i = H_{((i-1) \bmod n)+1}$ for $i = 1, \dots, \mu$. The theorem follows from iterated application of the claim for the case $1 \leq \mu \leq n$. \square

Following the proof of Theorem 2.1 we can derive.

COROLLARY 2.2. *For every complete graph $G = (V, E)$ with $|V| = n$ and every $k \geq 1$ there is a population of size $\mu = kn$ such that each edge is used exactly $2k$ times.*

Moreover, for the case of $1 \leq \mu \leq \lfloor \frac{n}{2} \rfloor$ it follows that a set of μ edge-disjoint STs is possible. From now on this will be the case of interest unless told otherwise.

COROLLARY 2.3. *For every complete graph $G = (V, E)$ with $|V| = n$ and every $1 \leq \mu \leq \lfloor \frac{n}{2} \rfloor$ there is a population P of μ edge-disjoint spanning trees with maximum diversity $D_o(P) = \mu(\mu-1)(n-1)$.*

3 STRUCTURAL PROPERTIES AND RUNTIME RESULTS

We consider the easiest non-trivial case: $\mu = 2$. We study the performance of $(2+1)$ -EA (see Algorithm 1) on complete graphs with $n \geq 4$ nodes where all edges have the same cost. We are interested in the *expected number of function evaluations* until a D_o -maximal population is found for the first time; the most common measure in the time complexity analysis of randomized search heuristics [9]. The fitness function for the EA is the edge overlap which we aim to minimise (cf. Eq 1); recall that this is equivalent to maximising the overlap diversity.

Clearly, given two STs T_1, T_2 the fitness can be improved by replacing an overlap edge, i.e. and edge $e \in E(T_1) \cap E(T_2)$ with a *free* edge which is not used by neither T_1 and T_2 . It turns out that even for $\mu = 2$ a single one-edge exchange mutation may not be sufficient to reduce the overlap. Consider the case where T_1 and T_2 are star graphs and there is just a single overlap edge $\{u, v\}$ which links the center nodes u of T_1 and v of T_2 (see Fig. 2). In this case any alternative edge for $\{u, v\}$ in T_1 and T_2 is also in T_2 and T_1 . Hence, in this special constellation a fitness plateau is reached which cannot be surpassed by a single 1-EX move. However, two consecutive 1-EX moves are sufficient to surpass this fitness plateau: w.l.o.g. replace $\{u, v\}$ in T_1 with one of the other suitable edges. Now T_1 is no star graph anymore and an improving step for T_2 exists.

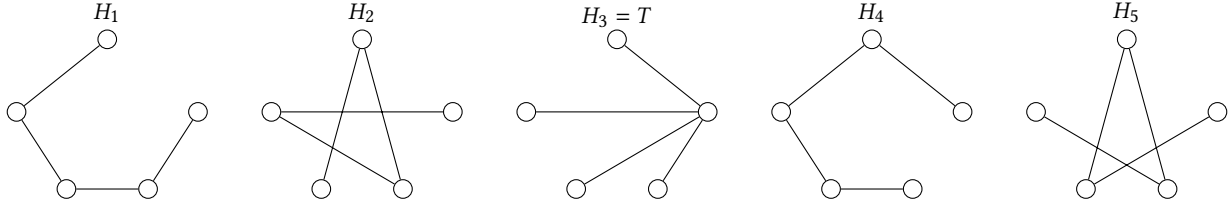


Figure 1: Illustration of set $H = \{H_1, \dots, H_n\}$ for $n = 5$ constructed in Theorem 2.1.

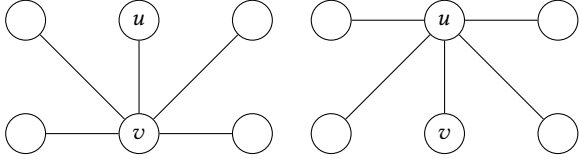


Figure 2: Two spanning trees of star graph type with center nodes v for T_1 (left) and u for T_2 (right).

In the following we show that in case of overlap $o(T_1, T_2) \geq 2$ there always exists a 1-EX step that strictly improves the diversity.

LEMMA 3.1. *On every complete graph with n nodes and population $P = \{T_1, T_2\}$, if $o(T_1, T_2) \geq 2$ then each edge $e \in E(T_1) \cap E(T_2)$ can be replaced with $k - 1$ other edges in either tree, such that the overlap reduces by one.*

PROOF. Let $k = o(T_1, T_2) \in \{2, \dots, n - 1\}$. Choose an arbitrary overlap edge $e \in (E(T_1) \cap E(T_2))$. W. l. o. g. set $T'_1 = T_1 \setminus \{e\}$. Since T_1 is a spanning tree, removing e destroys connectivity and decomposes T'_1 into a cut, i.e. two disjoint node sets C_1 and C_2 . It should be noted that 1-EX works differently: first insert an edge and remove an edge from the unique introduced cycle. However, for the sake of intuition we might think the other way around. In order to re-establish the spanning tree property in T' a cut-edge (an edge that has one end node in C_1 and the other in C_2) needs to be added. There are $|C_1| \cdot |C_2|$ such edges since G is complete. Now $|C_1| \cdot |C_2| \geq (n - 1)$. This happens if one node set has a single isolated node. Since e is a cut-edge there are $(n - 2)$ alternative cut-edges left. The overlap is k . In consequence at most $(n - 1) - k$ of those $(n - 2)$ alternatives are used by T_2 . Hence, the number of non-used cut edges is at least $(n - 2) - ((n - 1) - k) = k - 1 \geq 1$ since $k \geq 2$. Hence, we can always include one of those edges into T_1 and remove e which leads to a reduction of the overlap by one. \square

THEOREM 3.2. *On every complete graph with n nodes $(2 + 1)$ -EA with one-edge-exchange mutation, starting with two clones of an arbitrary spanning tree, needs expected time $O(mn \log n)$ to find a D_o -maximal population.*

PROOF. Let T_1 and T_2 be the two spanning trees. Let $k = o(T_1, T_2) \in \{0, \dots, n - 1\}$ with $k = n - 1$ after initialization. We define natural disjoint fitness-levels [9]

$$A_k = \{P = \{T_1, T_2\} \mid o(T_1, T_2) = k\},$$

i. e., the k -th fitness level contains all populations with overlap k and A_0 contains all populations of maximum D_o -diversity. In order

to leave A_k and transition into A_{k-1} or lower (note that we are minimizing overlap here), we need to reduce the overlap by at least one. For $k \geq 2$ by Lemma 3.1 we know that we can include at least $(k - 1)$ different edges into either T_1 or T_2 to produce a cycle with a cut edge on it. The probability for this event is at least $(k - 1)/m$. The resulting cycle has length at most n . Thus, the probability to select the overlap edge is at least $1/n$. By folklore waiting time arguments the expected waiting time for such an improving step is at most $(mn)/(k - 1)$. Summing up we obtain the following upper bound until we reach A_1 :

$$\sum_{k=2}^{n-1} \frac{mn}{k-1} = mn \sum_{k=1}^{n-2} \frac{1}{k} = mn \cdot H_{n-2} = O(mn \log n).$$

Here, $H_n = \sum_{i=1}^n (1/i) = \log(n) + \Theta(1)$ is the Harmonic sum. The last step is to reach A_0 from A_1 . This may be possible in a single step, but there might be a problem if T_1 and T_2 are star graphs where the overlap edge links the center nodes. This situation can be resolved by two consecutive mutations with an escape-state when one of the trees is no star graph anymore. In this situation a single one-edge exchange in the other tree can get rid of the last overlap. There is a fair random walk on the states of the fitness plateau which ends in the escape state in expected time $O(mn)$ as each step happens with probability at least $1/(mn)$. Eventually, eliminating the last overlap requires another $O(mn)$ steps in expectation since the overlap edge and an adequate replacement edge have to be selected; again, both events happen with probability at least $1/(mn)$. In sum the expected time to reach A_0 remains $O(mn \log n)$. \square

Theorem 3.2 states that $(\mu + 1)$ -EA is capable of finding a population of maximum diversity in polynomial time for $\mu = 2$; a satisfying result. However, the probabilities derived in Lemma 3.1 are rather pessimistic. The number of cut-edges which link the two connected components C_1 and C_2 when an edge is dropped from one tree is at least $(n - 1)$. However, if $|C_1| = |C_2| = (n/2)$ (n even), there are $n^2/4 = \Theta(n^2)$ many cut edges. In consequence the probability to reduce the overlap is much higher. This could indicate that on average the running time is lower. We now turn the focus to experimental investigations with a broader range of μ -values in the following sections.

4 EXPERIMENTAL STUDY

We presented theoretical insights into the expected runtime of $(\mu + 1)$ -EA for the case $\mu = 2$. In this section we perform a series of experiments for general $2 \leq \mu \leq \lfloor \frac{n}{2} \rfloor$. We start with the unconstrained case, i.e., spanning trees are not required to meet a

Table 1: Comparison in terms of diversity (D_o), the mean number of iterations (mean) until the algorithm terminated, the standard deviation of iterations (std) and results of Wilcoxon-Mann-Whitney tests (stat). The data is split by the sampling strategy (Uniform or Poisson). Here, X^+ means that the number of iterations is significantly better, i.e. lower, for the respective strategy. Lowest mean values per row are highlighted .

n	μ	Uniform[1] (1)				Uniform[2] (2)				Uniform[3] (3)				Poisson (4)			
		D_o	mean	std	stat	D_o	mean	std	stat	D_o	mean	std	stat	D_o	mean	std	stat
50	2	100.00	488.70	164.8		100.00	333.97	144.0	1 ⁺	100.00	243.93	104.5	1 ⁺ , 2 ⁺	100.00	198.67	85.2	1 ⁺ , 2 ⁺ , 3 ⁺
	10	100.00	7153.43	2389.4		100.00	5289.17	1469.0	1 ⁺	100.00	4660.60	1514.0	1 ⁺ , 2 ⁺	100.00	3741.43	1185.4	1 ⁺ , 2 ⁺ , 3 ⁺
	25	99.80	62500.00	0.0		99.79	62500.00	0.0		99.77	62500.00	0.0		99.74	62500.00	0.0	
100	2	100.00	1624.93	510.2		100.00	1072.43	429.8	1 ⁺	100.00	642.27	283.8	1 ⁺ , 2 ⁺	100.00	578.70	255.8	1 ⁺ , 2 ⁺
	10	100.00	16836.80	3096.7		100.00	11373.37	3580.1	1 ⁺	100.00	8120.70	2252.3	1 ⁺ , 2 ⁺	100.00	7603.67	2459.1	1 ⁺ , 2 ⁺
	25	100.00	73498.87	18139.3		100.00	59131.93	15166.8	1 ⁺	100.00	58792.83	19515.9	1 ⁺	100.00	52915.17	18007.0	1 ⁺ , 2 ⁺
	50	99.93	500000.00	0.0		99.92	500000.00	0.0		99.91	500000.00	0.0		99.89	500000.00	0.0	
200	2	100.00	4366.17	1200.9		100.00	2757.70	999.9	1 ⁺	100.00	1981.87	809.1	1 ⁺ , 2 ⁺	100.00	1523.53	645.6	1 ⁺ , 2 ⁺ , 3 ⁺
	10	100.00	45564.87	11035.6		100.00	30632.00	5848.0	1 ⁺	100.00	19364.20	3210.8	1 ⁺ , 2 ⁺	100.00	16428.40	3630.1	1 ⁺ , 2 ⁺ , 3 ⁺
	25	100.00	153040.03	30063.4		100.00	114716.33	24364.6	1 ⁺	100.00	86105.70	25137.5	1 ⁺ , 2 ⁺	100.00	74420.97	19746.9	1 ⁺ , 2 ⁺ , 3 ⁺
	50	100.00	448873.77	56975.2		100.00	422085.90	97636.7	1 ⁺	100.00	396190.57	60480.1	1 ⁺	100.00	360588.67	92265.0	1 ⁺ , 2 ⁺ , 3 ⁺
	100	99.98	4000000.00	0.0		99.97	4000000.00	0.0		99.96	4000000.00	0.0		99.95	4000000.00	0.0	
400	2	100.00	15081.70	4080.7		100.00	9765.10	3337.4	1 ⁺	100.00	5980.50	1873.9	1 ⁺ , 2 ⁺	100.00	5139.60	1834.1	1 ⁺ , 2 ⁺ , 3 ⁺
	10	100.00	126942.27	29267.6		100.00	80306.63	17046.5	1 ⁺	100.00	56559.83	10089.8	1 ⁺ , 2 ⁺	100.00	42918.23	8707.2	1 ⁺ , 2 ⁺ , 3 ⁺
	25	100.00	361950.37	43469.4		100.00	261919.33	46744.9	1 ⁺	100.00	189700.70	31737.1	1 ⁺ , 2 ⁺	100.00	152788.93	27460.6	1 ⁺ , 2 ⁺ , 3 ⁺
	50	100.00	1017406.93	216403.9		100.00	731349.90	96224.3	1 ⁺	100.00	588732.20	83189.7	1 ⁺ , 2 ⁺	100.00	543359.37	112333.1	1 ⁺ , 2 ⁺ , 3 ⁺
	100	100.00	3344395.80	377191.7		100.00	3019514.87	550293.3	1 ⁺	100.00	2827195.20	454872.3	1 ⁺	100.00	2541803.50	495036.2	1 ⁺ , 2 ⁺ , 3 ⁺

minimum quality. After that we consider the constrained case with different quality thresholds.

4.1 Reproducibility

All experiments have been conducted on a HPC cluster with Intel Xeon Gold 6140 18C 2.30GHz (Skylake) CPUs and 16GB RAM. Algorithm 1 was implemented in the statistical programming language R [28] in version 3.6.0 for rapid prototyping interfacing C++ for performance. Parallel job distribution/management was accomplished with the R package batchtools [18]. We also relied on the random number generator adopted by batchtools. Code and data are available in a public GitHub repository.²

4.2 Unconstrained Diversity Optimization

In the unconstrained setting we consider complete graphs $G = (V, E)$ with $n \in \{50, 100, 200, 400\}$, $m = n(n-1)/2$ and μ in the range between 2 and $\lfloor \frac{n}{2} \rfloor$. As our focus is on the algorithms ability to cope with the diversity part the considered setting adopts $c(e) = 1 \forall e \in E$. This way the check in line 4 of Algorithm 1 is always true. Note that for this setup the statement from Theorem 2.3 holds.

We are interested in the number of function evaluations required by $(\mu+1)$ -EA to reach a population of maximum diversity. Therefore, we adopt two stopping criteria: (1) stop when a maximum diversity

population is reached or (2) stop if the budget of μn^2 function evaluations (FE) is depleted. In addition, we study the effect of a single 1-EX, i.e. performing a single one-edge-exchange per mutation as stated in Algorithm 1 in comparison to possibly multiple subsequent 1-EX moves. Here, we consider the case where the number of swaps is sampled from $\{1, \dots, l\}$ for $l = 1, 2, 3$ uniformly at random (this is denoted by Uniform[l] in the following and includes the case of a single 1-EX move). In addition we consider the number of 1-EX moves being sampled from a Poisson distribution with rate $\lambda = 1$. Here, there is a strictly positive probability to exchange multiple edges which may be advantageous and necessary to leave local optima. For each $(n, \mu, \text{sampling})$ -combination we run $(\mu + 1)$ -EA 30 times for statistical soundness.

Table 1 shows a comparison in terms of population diversity D_o in percent as well as summary statistics for the number of iterations required. The results indicate that $(\mu + 1)$ -EA manages to find a μ -size population of maximum diversity for almost all considered scenarios within the limit of μn^2 function evaluations. Solely the case where $\mu = \lfloor \frac{n}{2} \rfloor$ seems tricky. Here, $(\mu + 1)$ -EA apparently gets trapped in local optima and is unable to escape (within the given FE-limit): all 30 runs consistently hit the imposed FE-limit. In addition we observe a near-constant advantage for multiple 1-EX moves per mutation. For relative low μ $(\mu + 1)$ -EA with Poisson sampling is up to three times as fast as its competitor and also shows lower standard deviation. The results are statistically significant as

²Code and data: <https://github.com/jakobbossek/GECCO2021-mst-diversity>.

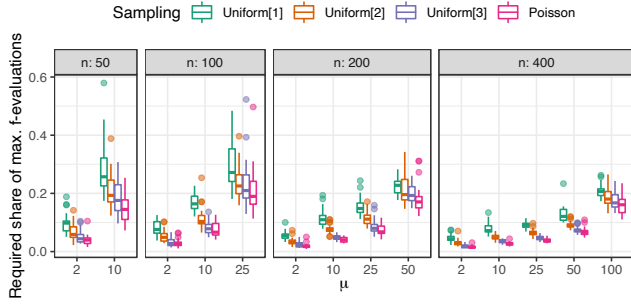


Figure 3: Distribution of the share of maximum function evaluations (μn^2) in the unconstrained case split by instance size n and population size μ .

indicated by the 1^+ values in the **stat** column; a plus symbol shows that the zero hypothesis of lower number of FEs for the competitor algorithm was rejected by the Wilcoxon-Mann-Whitney test at a significance level of 5%. This effect becomes weaker when μ approaches $\lfloor \frac{n}{2} \rfloor$. In this region it apparently becomes more beneficial – in the end of the optimization process – to produce mutants by single 1-EX moves. Figure 3 gives a visual impression of the number of function evaluations required as a share of the maximum FE-limit μn^2 . The box-plots confirm our observations from Table 2. In conclusion, this series of experiments verifies empirically that $(\mu + 1)$ -EA is able to efficiently evolve sets of edge-disjoint STs.

The focus of this paper is on the overlap-based diversity measure. However, in real-world applications, e.g. telecommunication networks, diversity with respect to additional structural features is desirable. To reduce the workload of single nodes in said networks, constraints on the maximum degree of the MST pose real-world requirements. In this context also MSTs with many leaves are advantageous. Furthermore, low-diameter³ MSTs enable fast communication since the number of hops is kept relatively low [14]. For MSTs on complete graphs the number of leaves, the maximum degree and the diameter can take $n - 2$ different values in $\{2, 3, \dots, n - 1\}$ where in all three cases n -vertex paths and star graphs pose the extreme cases; i.e. a star graph has $(n - 1)$ leaves, maximum degree $(n - 1)$ and diameter 2 while a path has diameter $n - 1$ with maximum degree 2 and 2 leaves. Table 2 shows mean and standard deviation of diversity (as percentage values) with respect to the maximum degree, the number of leaves and the diameter in the unconstrained case. Due to space limitations we show the statistics for the Poisson-sampling only. For these features diversity is measured as the fraction of distinct values in the final populations. E.g. maximum max-degree diversity is obtained if all trees have different maximum degree.⁴ The values in the table indicate a clear trend: with increasing μ the mean diversity values decrease. This trend is particularly strong for the maximum degree. Note that this is in line with Corollary 2.3 where the proof was constructive and the resulting population was a set of edge-disjoint paths all with maximum degree 2. The data

³The diameter of a graph is the length of its longest simple path.

⁴Note that in our setup $\mu \leq \lfloor \frac{n}{2} \rfloor < n - 2$ and hence maximum diversity could be possible. However, for μ close to the maximum value we actually do not know if this is achievable in theory.

Table 2: Diversity (in percent of the maximally achievable value) with respect to overlap diversity D_o , maximum degree (Max. degree), number of leafs (Leaf) and the diameter (Diameter) in the unconstrained setting.

n	μ	D_o		Max. degree		Leaf		Diameter	
		mean	std	mean	std	mean	std	mean	std
50	2	100.00	0.00	83.33	23.97	88.33	21.51	95.00	15.26
	10	100.00	0.00	30.67	6.40	61.67	11.17	65.67	11.65
	25	99.80	0.02	14.67	2.43	33.33	4.25	41.07	5.03
100	2	100.00	0.00	85.00	23.30	96.67	12.69	95.00	15.26
	10	100.00	0.00	30.67	7.40	66.33	9.64	76.33	9.99
	25	100.00	0.00	16.27	3.14	43.07	5.82	50.40	4.99
	50	99.93	0.00	8.87	1.14	26.07	3.34	32.07	3.46
200	2	100.00	0.00	75.00	25.43	95.00	15.26	96.67	12.69
	10	100.00	0.00	35.00	5.72	78.33	8.34	81.33	7.76
	25	100.00	0.00	15.20	2.44	53.73	5.91	60.13	6.52
	50	100.00	0.00	8.33	1.06	33.00	2.45	40.53	3.52
400	100	99.98	0.00	4.53	0.63	20.70	1.49	24.37	1.79
	2	100.00	0.00	83.33	23.97	95.00	15.26	95.00	15.26
	10	100.00	0.00	30.33	7.65	81.33	9.37	81.67	13.15
	25	100.00	0.00	16.13	2.46	61.47	6.68	66.67	6.83
400	50	100.00	0.00	8.87	1.14	43.13	3.43	47.47	4.81
	100	100.00	0.00	5.03	0.61	26.53	2.08	30.63	2.65

suggests, that in order to obtain higher diversity with respect to, e.g. maximum degree, one likely has to sacrifice overlap diversity. This is clearly a multi-objective problem and will be part of our future agenda.

5 CONSTRAINED DIVERSITY OPTIMIZATION

We now turn our focus to constrained diversity optimization on random Euclidean graphs where nodes are associated with point coordinates in the Euclidean plane. To this end we consider $n \in \{50, 100, 200, 400\}$, $\mu \in \{2, 10, 25, 50, 100, 200\}$ such that $\mu \leq \lfloor \frac{n}{2} \rfloor$ and $\alpha \in \{0.05, 0.10, 0.5, 1\}$. Consistent with the setup in the unconstrained case $(\mu + 1)$ -EA is run 30 times on each generated instance for at most μn^2 iterations. Table 3 shows the results following the format of Table 1. We show a subset of the results for Uniform[1] and Poisson sampling for $n \in \{50, 100\}$; the omitted data shows the same picture. As expected diversity in most cases increases with increasing α as there is more flexibility in choosing alternative edges (see Figure 4 for a superimposed visualisation of solutions).

In addition, a distinctive pattern with respect to the sampling strategies can be observed. For low α , performing single 1-EX moves seems advantageous. In contrast, performing multiple 1-EX moves mutation leads to better results in terms of overlap diversity for increasing α . This seems plausible and in par with our observations in Section 4: as α increases, so does the size of the feasible search space. Hence, stronger mutation may explore this space more thoroughly as multiple subsequent edge exchanges might be necessary in order to come up with a solution that in fact adheres to the quality constraint. However, this effect diminishes with μ approaching

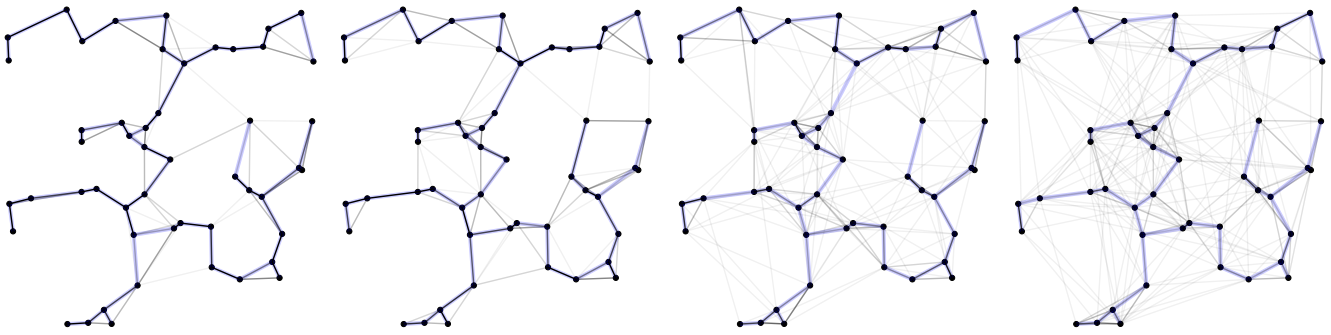


Figure 4: Visualised edge usage for an Euclidean instance with $n = 50$ nodes, $\mu = 10$ and $\alpha \in \{0.05, 0.10, 0.50, 1.00\}$ (from left to right). In the plots all μ spanning trees are superimposed. The darker an edge, the higher the overlap in the population. Omitted edges are not part of any solution. MST-edges are shown in blue.

Table 3: Comparison in terms of mean overlap diversity (mean), its standard deviation (std) and results of Wilcoxon-Mann-Whitney tests (stat). The data is split by the sampling strategy (Single or Poisson).

n	μ	α	Uniform[1] (1)			Poisson (2)		
			mean	std	stat	mean	std	stat
50	2	0.05	21.97	6.17	2 ⁺	19.12	3.96	
		0.10	26.73	4.49	2 ⁺	24.56	4.59	
		0.50	47.14	4.92		66.80	5.44	1 ⁺
		1.00	62.86	5.98		91.29	4.55	1 ⁺
	10	0.05	28.47	1.41	2 ⁺	22.35	1.29	
		0.10	36.69	1.41	2 ⁺	29.97	1.61	
		0.50	63.60	1.97		64.57	1.06	1 ⁺
		1.00	75.34	2.48		83.56	0.64	1 ⁺
	25	0.05	29.15	1.02	2 ⁺	26.07	1.25	
		0.10	38.28	0.71	2 ⁺	35.13	1.04	
		0.50	68.23	0.81	2 ⁺	67.43	0.56	
		1.00	81.56	0.66		83.28	0.38	1 ⁺
2	0.05	11.41	2.85		11.68	2.58		
	0.10	15.89	4.48		19.46	3.59	1 ⁺	
	0.50	31.14	4.96		58.86	3.44	1 ⁺	
	1.00	47.30	4.26		85.79	2.69	1 ⁺	
100	10	0.05	24.68	1.06	2 ⁺	15.92	1.47	
		0.10	33.87	1.33	2 ⁺	24.37	1.50	
		0.50	59.10	2.51		59.07	1.05	
		1.00	70.94	3.33		80.41	0.72	1 ⁺
	25	0.05	26.24	0.61	2 ⁺	20.53	0.92	
		0.10	35.98	0.71	2 ⁺	29.57	0.96	
		0.50	66.55	0.68	2 ⁺	63.75	0.57	
		1.00	79.78	0.69		81.13	0.45	1 ⁺
	50	0.05	26.69	0.44	2 ⁺	22.90	0.77	
		0.10	36.64	0.40	2 ⁺	32.16	0.64	
		0.50	67.86	0.35	2 ⁺	66.30	0.43	
		1.00	81.74	0.46		82.17	0.20	1 ⁺

Table 4: Diversity (in percent of the maximally achievable value) with respect to overlap diversity D_o , maximum degree (Max. degree), number of leafs (Leaf) and the diameter (Diameter) in the constrained setting.

n	μ	α	D_o		Max. degree		Leaf		Diameter	
			mean	std	mean	std	mean	std	mean	std
50	2	0.05	19.12	3.96	75.00	25.43	88.33	21.51	95.00	15.26
		0.10	24.56	4.59	75.00	25.43	95.00	15.26	96.67	12.69
		0.50	66.80	5.44	75.00	25.43	93.33	17.29	96.67	12.69
		1.00	91.29	4.55	80.00	24.91	88.33	21.51	88.33	21.51
	10	0.05	22.35	1.29	18.33	5.92	42.00	9.97	43.00	11.49
		0.10	29.97	1.61	22.67	7.85	44.00	9.32	55.67	11.35
		0.50	64.57	1.06	27.33	6.40	57.67	8.98	66.67	10.93
		1.00	83.56	0.64	29.67	6.69	56.67	9.22	66.33	9.28
	25	0.05	26.07	1.25	9.60	1.99	20.13	4.26	23.47	5.22
		0.10	35.13	1.04	10.93	2.08	23.60	4.74	26.53	4.87
		0.50	67.43	0.56	12.93	3.10	32.93	4.89	36.67	3.65
		1.00	83.28	0.38	14.27	2.50	34.80	5.37	39.87	5.51
100	2	0.05	11.68	2.58	65.00	23.30	90.00	20.34	96.67	12.69
		0.10	19.46	3.59	65.00	23.30	96.67	12.69	91.67	18.95
		0.50	58.86	3.44	81.67	24.51	96.67	12.69	98.33	9.13
		1.00	85.79	2.69	83.33	23.97	91.67	18.95	91.67	18.95
	10	0.05	15.92	1.47	17.00	5.96	44.33	10.40	46.00	11.02
		0.10	24.37	1.50	20.33	6.15	56.00	8.14	62.33	10.40
		0.50	59.07	1.05	26.00	5.63	61.00	11.25	80.00	7.88
		1.00	80.41	0.72	32.00	6.64	67.00	11.19	75.33	10.08
	25	0.05	20.53	0.92	7.73	3.14	23.87	4.52	27.73	4.32
		0.10	29.57	0.96	9.07	2.33	27.47	5.82	33.60	6.36
		0.50	63.75	0.57	12.80	1.63	40.27	5.94	53.20	5.96
		1.00	81.13	0.45	14.67	2.64	41.87	4.67	52.00	6.73
50	0.05	22.90	0.77	4.13	1.48	14.47	2.91	16.40	2.99	
	0.10	32.16	0.64	4.47	1.01	17.33	3.08	21.20	3.81	
	0.50	66.30	0.43	6.53	0.90	24.40	3.17	31.47	2.83	
	1.00	82.17	0.20	7.60	1.33	26.20	3.17	33.67	2.41	

$\lfloor \frac{n}{2} \rfloor$. We see that single mutation steps become more effective and the gap between single step and multi step mutation decreases even in the case where multi-step mutation is still stronger.

Table 4 explores diversity with respect to the maximum degree, the number of leafs and the diameter for the constrained setting. For fixed α diversity with respect to all three features decreases with increasing μ showing an inverse trend in comparison with overlap diversity. In line with the unconstrained case the maximum degree diversity is consistently lowest reaching below 15% for $\mu = \lfloor \frac{n}{2} \rfloor$. Keeping μ fixed we see a trend towards higher diversity values for increasing α . This seems plausible as for low α the final population differs in only few edges (high overlap) and hence the STs are also more likely to be structurally similar.

6 CONCLUSION

We studied the minimum spanning tree problem in the context of evolutionary diversity optimization for the first time. Here, the goal is to evolve a population of high quality solutions which all satisfy minimum quality requirements. We studied a baseline $(\mu + 1)$ evolutionary algorithm with a diversity measure based on pairwise edge overlap of solutions. Runtime complexity results show that a $(2 + 1)$ -EA is able to find two maximally diverse spanning trees in expected time $O(n^3 \log n)$ in the unconstrained setting where n is the number of nodes; one of the first runtime results for evolutionary diversity optimization on well-known combinatorial optimization problems. Complementary experiments suggest that $(\mu + 1)$ -EA with μ up to $\lfloor \frac{n}{2} \rfloor$ can efficiently evolve diverse populations of edge-disjoint spanning trees in the unconstrained and the constrained case where in the latter a relaxation of the quality constrained allows for higher diversity. However, increasing population size leads to less diversity with of solutions with respect to other desired tree features: the maximum degree, the number of leafs and the diameter.

Future research endeavors will focus on improving and generalizing the theoretical runtime bounds to strengthen the still rare theoretical foundations of evolutionary diversity optimization on combinatorial optimization problems. Here, in particular an extension towards the constrained case seems interesting and challenging. In addition, multi-objective approaches which aim to simultaneously maximise several diversity measures appear promising.

ACKNOWLEDGMENT

This work was supported by the Australian Research Council through grant DP190103894.

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 (GECCO '17)*, Peter A. N. Bosman (Ed.). ACM, 171–178. <https://doi.org/10.1145/3071178.3071342>

[2] Brian Alspach, Jean-Claude Bermond, and Dominique Sotteau. 1990. *Decomposition into cycles I: Hamilton decompositions*. Springer Netherlands, 9–18.

[3] Jakob Bossek, Christian Grimme, and Frank Neumann. 2019. On the benefits of biased edge-exchange mutation for the multi-criteria spanning tree problem. In *Proceedings of the 2019 Genetic and Evolutionary Computation Conference (GECCO '19)*. ACM Press, New York, New York, USA, 516–523. <https://doi.org/10.1145/3321707.3321818>

[4] Jakob Bossek, Pascal Kerschke, Aneta Neumann, Markus Wagner, Frank Neumann, and Heike Trautmann. 2019. Evolving diverse TSP instances by means of novel and creative mutation operators. In *Proceedings of the 15th ACM/SIGEVO*

Conference on Foundations of Genetic Algorithms (FOGA XV). ACM Press, New York, New York, USA, 58–71. <https://doi.org/10.1145/3299904.3340307>

[5] Gary Chartrand and Linda Lesniak. 1986. *Graphs & Digraphs (2nd Ed.)*. Wadsworth Publ. Co., USA.

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2001. *Introduction to Algorithms* (2nd ed.). The MIT Press.

[7] Duc-Cuong Dang, Thomas Jansen, and Per Kristian Lehre. 2017. Populations Can Be Essential in Tracking Dynamic Optima. *Algorithmica* 78, 2 (2017), 660–680.

[8] Anh Viet Do, Jakob Bossek, Aneta Neumann, and Frank Neumann. 2020. Evolving Diverse Sets of Tours for the Travelling Salesperson Problem. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 681–689. <https://doi.org/10.1145/3377930.3389844>

[9] Doerr, B., Neumann, F. (Eds.). 2020. *Theory of Evolutionary Computation – Recent Developments in Discrete Optimization*. Springer. <https://doi.org/10.1007/978-3-030-29414-4>

[10] Tobias Friedrich and Frank Neumann. 2015. Maximizing Submodular Functions under Matroid Constraints by Evolutionary Algorithms. *Evolutionary Computation* 23, 4 (2015), 543–558.

[11] Wanru Gao, Samadhi Nallaperuma, and Frank Neumann. 2016. Feature-based diversity optimization for problem instance classification. In *Parallel Problem Solving from Nature, PPSN (LNCS)*, Vol. 9921. Springer, 869–879. https://doi.org/10.1007/978-3-319-45823-6_81

[12] Wanru Gao and Frank Neumann. 2014. Runtime analysis for maximizing population diversity in single-objective optimization. In *Proceedings of the 2014 Genetic and Evolutionary Computation Conference (GECCO '14)*. ACM, 777–784.

[13] Wanru Gao, Mojgan Pourhassan, and Frank Neumann. 2015. Runtime Analysis of Evolutionary Diversity Optimization and the Vertex Cover Problem. In *Proceedings of the 2015 Genetic and Evolutionary Computation Conference (Companion) (GECCO '15)*. ACM, 1395–1396.

[14] Luis Gouveia and Thomas L. Magnanti. 2003. Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. *Networks* 41, 3 (2003), 159–173. <https://doi.org/10.1002/net.10069> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.10069>

[15] Thomas Jansen. 2013. *Analyzing Evolutionary Algorithms - The Computer Science Perspective*. Springer. 1–236 pages.

[16] Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. 2019. Automated Algorithm Selection: Survey and Perspectives. *Evolutionary Computation* 27, 1 (2019), 3–45. https://doi.org/10.1162/evco_a_00242

[17] Joseph B. Kruskal. 1956. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proc. Amer. Math. Soc.* 7, 1 (1956), 48–50.

[18] Michel Lang, Bernd Bischl, and Dirk Surmann. 2017. batchtools: Tools for R to work on batch systems. *The Journal of Open Source Software* 2, 10 (feb 2017). <https://doi.org/10.21105/joss.00135>

[19] Joel Lehman and Kenneth O Stanley. 2011. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 2011 Genetic and Evolutionary Computation Conference*. ACM, 211–218.

[20] Aneta Neumann, Wanru Gao, Carola Doerr, Frank Neumann, and Markus Wagner. 2018. Discrepancy-based evolutionary diversity optimization. In *Proceedings of the 2018 Genetic and Evolutionary Computation Conference (GECCO '18)*. ACM, 991–998. <https://doi.org/10.1145/3205455.3205532>

[21] Aneta Neumann, Wanru Gao, Markus Wagner, and Frank Neumann. 2019. Evolutionary diversity optimization using multi-objective indicators. In *Proceedings of the 2019 Genetic and Evolutionary Computation Conference (GECCO '19)*. ACM, 837–845. <https://doi.org/10.1145/3321707.3321796>

[22] Aneta Neumann and Frank Neumann. 2020. Optimising Monotone Chance-Constrained Submodular Functions Using Evolutionary Multi-objective Algorithms. In *Proceedings of the Parallel Problem Solving from Nature, PPSN 2020 (Lecture Notes in Computer Science)*, Vol. 12269. Springer, 404–417.

[23] Frank Neumann and Carsten Witt. 2010. *Bioinspired Computation in Combinatorial Optimization*. Springer. <https://doi.org/10.1007/978-3-642-16544-3>

[24] R. C. Prim. 1957. Shortest Connection Networks And Some Generalizations. *Bell System Technical Journal* 36, 6 (1957), 1389–1401. <https://doi.org/10.1002/j.1538-7305.1957.tb01515.x>

[25] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. 2016. An Extended Study of Quality Diversity Algorithms. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. ACM, 19–20.

[26] Chao Qian, Jing-Cheng Shi, Yang Yu, and Ke Tang. 2017. On Subset Selection with General Cost Constraints. In *Proceedings of the 2017 International Joint Conference on Artificial Intelligence (IJCAI '19)*. 2613–2619.

[27] Chao Qian, Yang Yu, and Zhi-Hua Zhou. 2015. Subset Selection by Pareto Optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS 2015*. 1774–1782.

[28] R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>

[29] Günther R Raidl, Gabriele Koller, and Bryant A Julstrom. 2006. Biased Mutation Operators for Subgraph-Selection Problems. *IEEE Transactions on Evolutionary*

- Computation* 10, 2 (2006), 145–156.
- [30] Vahid Roostapour, Aneta Neumann, Frank Neumann, and Tobias Friedrich. 2019. Pareto Optimization for Subset Selection with Dynamic Cost Constraints. In *Proceedings of the 2019 AAAI Conference on Artificial Intelligence (AAAI '19)*. AAAI Press, 2354–2361.
- [31] Tobias Storch. 2008. On the Choice of the Parent Population Size. *Evolutionary Computation* 16, 4 (2008), 557–578.
- [32] Tamara Ulrich and Lothar Thiele. 2011. Maximizing population diversity in single-objective optimization. In *Proceedings of the 2011 Genetic and Evolutionary Computation Conference (GECCO '11)*. ACM, 641–648.