On the Benefits of Biased Edge-Exchange Mutation for the Multi-Criteria Spanning Tree Problem

Jakob Bossek
Dept. of Information Systems
University of Münster, Germany
bossek@uni-muenster.de

Christian Grimme
Dept. of Information Systems
University of Münster, Germany
christian.grimme@uni-muenster.de

Frank Neumann
School of Computer Science
The University of Adelaide, Australia
frank.neumann@adelaide.edu.au

ABSTRACT

Research has shown that for many single-objective graph problems where optimum solutions are composed of low weight sub-graphs, such as the minimum spanning tree problem (MST), mutation operators favoring low weight edges show superior performance. Intuitively, similar observations should hold for multi-criteria variants of such problems. In this work, we focus on the multi-criteria MST problem. A thorough experimental study is conducted where we estimate the probability of edges being part of non-dominated spanning trees as a function of the edges' non-domination level or domination count, respectively. Building on gained insights, we propose several biased one-edge-exchange mutation operators that differ in the used edge-selection probability distribution (biased towards edges of low rank). Our empirical analysis shows that among different graph types (dense and sparse) and edge weight types (both uniformly random and combinations of Euclidean and uniformly random) biased edge-selection strategies perform superior in contrast to the baseline uniform edge-selection. Our findings are in particular strong for dense graphs.

CCS CONCEPTS

• **Applied computing → Multi-criterion optimization and decision-making**;

KEYWORDS

Multi-objective optimization, Combinatorial optimization, Minimum spanning tree, Biased mutation

ACM Reference Format:
Jakob Bossek, Christian Grimme, and Frank Neumann. 2019. On the Benefits of Biased Edge-Exchange Mutation for the Multi-Criteria Spanning Tree Problem. In *Genetic and Evolutionary Computation Conference (GECCO '19), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3321707.3321818

GECCO '19, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6111-8/19/07…$15.00
https://doi.org/10.1145/3321707.3321818

1 INTRODUCTION

Evolutionary algorithms (EAs) have been shown to be successful problem solvers in a wide range of application areas such as engineering, design, and manufacturing [8, 9]. They are often applied to problems where there is not enough knowledge on the problem at hand or where the problems are too complex to be solved by traditional optimization methods such as dynamic programming or mixed integer programming. Evolutionary algorithms have a huge success story within the area of multi-objective optimization. As EAs work with a set of solutions called a population, they are extremely well suited to compute trade-offs with respect to a given set of objective functions. Various types of evolutionary multi-objective algorithms (EMOAs) have been developed and applied over the last 25 years (see e. g., [10]). When using an evolutionary algorithm for a problem where there is some domain knowledge, such knowledge should be incorporated to make the algorithm more efficient. Different studies show that problem specific operators can lead to a significant improvement in terms of the quality of solutions obtained and the runtime required to do so (see, e. g., [13]).

In this paper, we design problem specific mutation operators for the multi-objective minimum spanning problem. This $\mathcal{NP}$-hard multi-objective optimization problem has obtained significant attention in the traditional operations research [12, 17] and evolutionary computation literature [6, 14, 19]. It generalizes the classical minimum spanning tree problem which can be solved in polynomial time by well-known algorithms such as Kruskal and Prim to more than one weight function. Spanning tree problems as abstractions of various network design problems play a crucial role in the area of optimization. The classical minimum spanning tree problem gets $\mathcal{NP}$-hard when imposing additional restrictions such as a diameter or degree bound that a solution has to meet. For the unconstrained and the degree-bounded minimum spanning problem, Raidl et al. [16] have designed problem specific edge-exchange operators that prefer edges of low weight to be included in a solution. Specifically, they rank edges according to weight and assign probabilities for including an edge as part of their mutation operator depending on the rank. More recently, Bossek and Grimme [5] have introduced a problem-tailored mutation operator for the multi-objective minimum spanning tree problems. Their algorithm selects random connected sub-graphs of a candidate solutions and replaces those with an optimal sub-tree with respect to a random weighted sum scalarization.

We present different approaches of generalizing the biased mutation approach of Raidl et al. [16] for single-objective spanning tree problems to the multi-objective case. The idea is to establish a ranking on the edges of the given graph in order to design such rank-based mutation operators. It should be noted that we expect our approach not only to be applicable to the multi-objective minimum spanning tree problem, but also to other problems where the objective function values are given by the sum of the chosen components of the problems. This includes for example the multi-objective Traveling Salesperson Problem where each edge has more than one weight and the trade-offs with respect to the different weight functions have to be computed.

The remainder of this work is structured as follows: In Section 2 we formulate the combinatorial problem under consideration and motivate our main research question. Section 3 introduces ranking schemes for edges in the multi-criteria case and conducts a probability estimation study. On top of this, we introduce several biased edge-selection strategies in Section 4 and perform a comparative study in Section 5. Section 6 concludes the work and gives pointers to promising future research directions.

2 RESEARCH QUESTION

Let $G = (V, E, c)$ be a connected graph with edge-weights $c(e) = (c_1(e), \ldots, c_o(e)) \, \forall e \in E$ where $c : E \to \mathbb{R}^o$ is a vector-valued function which assigns o weights to each edge of G. We abbreviate $n = |V|, m = |E|$, and write $e \in G$ to indicate that an edge is in the edge set of a graph. With slight abuse of notation we overload the cost function and denote the costs $c(G)$ of G as the component-wise sum of its edge weights, i. e., $c_i(G) := \sum_{e \in E} c_i(e)$ for $i \in \{1, \ldots, o\}$. A connected, acyclic subgraph $T = (V, E_T)$ of G with $E_T \subseteq E$ is termed a *spanning tree* of G. Let $\mathcal{T}$ denote the set of all spanning trees of G. The *multi-criteria minimum spanning tree* problem (mcMST) is to locate the set of optimal compromise solutions minimizing all cost functions simultaneously:

$$\min_{T \in \mathcal{T}} c(T) = (c_1(T), \ldots, c_o(T)) \, .$$

The concept of *Pareto-dominance* is adopted throughout the paper to define optimality: for $T_1, T_2 \in \mathcal{T}$ we say that T_1 *dominates* T_2, $T_1 \preceq T_2$ in formula, if $c_i(T_1) \leq c_i(T_2)$ for all $i \in \{1, \ldots, o\}$ and there is at least one objective $j \in \{1, \ldots, o\}$ with $c_j(T_1) < c_j(T_2)$. If there is no spanning tree which dominates T_1 it is termed *non-dominated* or *efficient*. Hence, considering the mcMST, we strive to identify the set of all non-dominated spanning trees $PS = \{T \in \mathcal{T} \mid \nexists T' \in \mathcal{T} \text{ with } T' \preceq T\}$, called *Pareto-set*, and its image in the objective space $c(PS) = \{c(T) \mid T \in \text{PS}\}$, called *Pareto-front*, respectively.

Note that for $o = 1$ the problem is easy in terms of computational complexity. Well-known algorithms, e. g., by Kruskal [15], find single-objective MSTs in polynomial time. When multiple objectives are considered, the optimization problem turns $\mathcal{NP}$-hard [17] and suffers from intractability.[1]

When the mcMST is tackled by means of an evolutionary algorithm a natural, plain and simple mutation operator is the *one-edge-exchange operator*. Given a feasible candidate solution $T \in \mathcal{T}$ it selects an edge $e \in E$ at random for inclusion into T, which yields $T' = T \cup \{e\}$. If $T' = T$ we are done. Otherwise, there is exactly one cycle in T'. Dropping an edge f of this cycle at random, we end up with another feasible candidate solution $T' \setminus \{f\}$. Usually, the underlying probability distribution for edge-selection is uniform, i. e., each edge is selected with probability $p(e) = 1/m$. Our main research question is: *Is it possible to alter/bias the probability distribution such that edges of "low-weight" are favored and convergence to the Pareto-front is accelerated?* We approach this question systematically in the following sections.

[1]Due to Cayley's theorem [7] the set $\mathcal{T}$ has cardinality n^{n-2}. Ruzika and Hamacher [17] provide an example where all $|\mathcal{T}|$ spanning trees are efficient.

3 RANKING OF EDGES IN THE MULTI-CRITERIA CASE

In their seminal paper, Raidl et al. [16] derived rank based edge-selection probabilities for edges to be inserted into candidate solutions during an EA optimization for single-objective MST problem among others. In preliminary experiments they empirically analyzed the probability with which edges appear in optimum spanning trees as a function of the edges' rank. The study reveals that low-ranked (low-weight) edges are more likely to be members of optimum solutions. It is a legitimate assumption, that in the multi-criteria scenario with $o \geq 2$ objectives, non-dominated spanning trees are mainly composed of "low-ranked" edges as well.[2] This assumption is theoretically founded, since each sub-tree of an efficient spanning tree is efficient (see [17], i. e., non-dominated among all such sub-trees.). However, since there is no total order in the

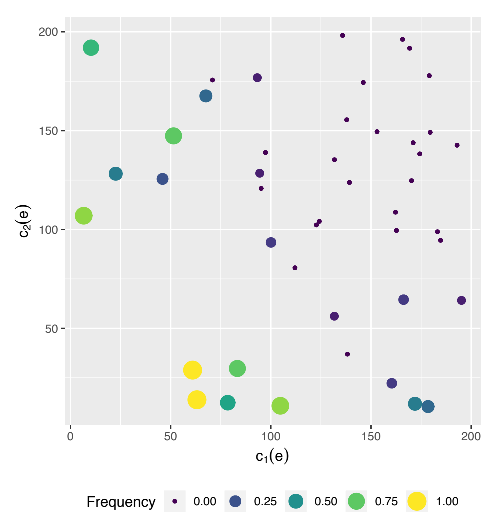

Figure 1: Scatterplot of edge weights for an exemplary instance of a complete graph with two random weights per edge. The point size and color indicates the fraction of efficient spanning trees the corresponding edge is part of.

multi-criteria case, we need to come up with an appropriate definition of the rank in a first step. As a starting point consider Figure 1, which shows a scatterplot of edge weights of a complete graph with $n = 10$ nodes and two uniform random weights. The point size and

[2]Note that sometimes the inclusion of large-weight edges is unavoidable. Consider the case of a bridge-edge, i. e., an edge whose deletion would cause the graph to be not connected anymore. Regardless of its weight and the definition of rank, this edge is part of any (non-dominated) spanning tree.

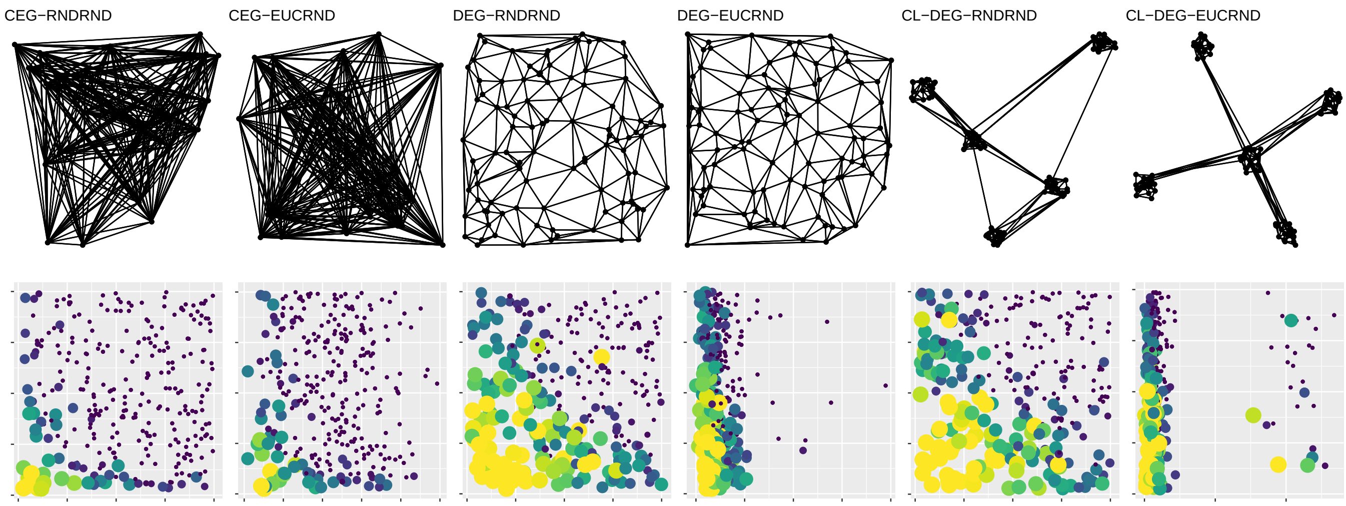

Figure 2: Exemplary graphs of each instance group considered in our study. The plots show the graph topology (top row) and a scatterplot of the corresponding edge weights (bottom row). The color and size of points in the latter plots indicates the relative share of (supported) efficient trees it is part of (the bigger/brighter the more frequent).

color indicates the share

$$s(e) = \frac{|\{T = (V, E_T) \in PS \mid e \in E_T\}|}{|PS|} \qquad (1)$$

of non-dominated spanning trees the corresponding edge is incorporated in. Obviously, edges which are dominated by only few other edges or are located on low non-domination levels are predominant building blocks of efficient spanning trees. This observation motivates the following two rank definitions for edges of a weighted graph $G = (V, E, c)$ adopted in this paper:

- **Non-domination level based** In this approach, edges are categorized into non-domination levels. The first level L_1 consists of all non-dominated edges of G. The second level is build up of all non-dominated edges of $E \setminus L_1$ and so on. Given a sequence of levels $L_1, \ldots, L_{l_{\max}}$, with $l_{\max}$ being the maximum level, we say that $e \in E$ has rank $r(e) = i$ if $e \in L_i$.
- **Domination count based** In this definition the rank of an edge is defined as $r(e) = |\{e' \in E \mid c(e') \preceq c(e)\}|$, i. e., the number of edges it is dominated by. In contrast to the non-domination level based formulation this rank definition is more fine-grained and exhibits a larger set of value on non-degenerate graphs.

We use the notion "rank" synonym to non-domination level or domination count based rank respectively in the remainder of this paper.

3.1 Estimation of rank probabilities

It is reasonable to assume that rank probabilities depend on the type of edge weights (purely random or a combination of geometric and random), graph topology (random or clustered) and/or density $2m/(n(n-1)) \in [0, 1]$. To account for this we considered three different graph topologies:

- **CEG** Nodes are placed uniformly at random in the Euclidean plane $[0, 100]^2$ and all pairwise edges exist (Complete Edge Generation).
- **DEG** Again, nodes are placed at random in the Euclidean plane. The interconnection of nodes is based on the Delauney triangulation (Delauney Edge Generation) of the point coordinates [11]. In contrast to the CEG-type these graphs are rather sparse with $m = \Theta(n)$.
- **CL-DEG** Here, the node placement is two-fold: 1) $k = 5$ cluster centers are placed in $[0, 100]^2$ by means of a Latin-Hypercube-Sample [18]. Next, each n/k nodes are placed uniformly at random around the cluster centers. Inner-cluster edges are based on a Delauney triangulation (Clustered Delauney Edge Generation). Edges between clusters are generated by considering the complete graph spanned by each $l = 3$ randomly selected nodes from each cluster and sequential MST single-objective MST computation to ensure the resulting graph to be connected.

Edge weights $c_1(e)$ and $c_2(e)$ are either both real numbers chosen independently at random from a uniform distribution $\mathcal{U}[5, 200]$ (denoted as RNDRND in the following) or $c_1(e)$ is based on the Euclidean distance of the point coordinates and $c_2(e)$ stems from a $\mathcal{U}[5, 200]$-distribution (EUCRND). The R software `grapherator` [3] was used for instance generation.

Combining both aspects results in the six graph types CEG-RNDRND, CEG-ECURND, DEG-RNDRND, DEG-EUCRND, CL-DEG-RNDRND and CL-DEG-EUCRND. Embeddings in the Euclidean plane are depicted alongside scatterplots of edge weights in Figure 2 for each one instance of each group. Glancing at the scatter-plots in the bottom row, we observe a strong tendency towards low-rank edges in particular for complete graphs again. For the more sparse graphs it seems that even mid-rank edges are frequent members of efficient spanning-trees.

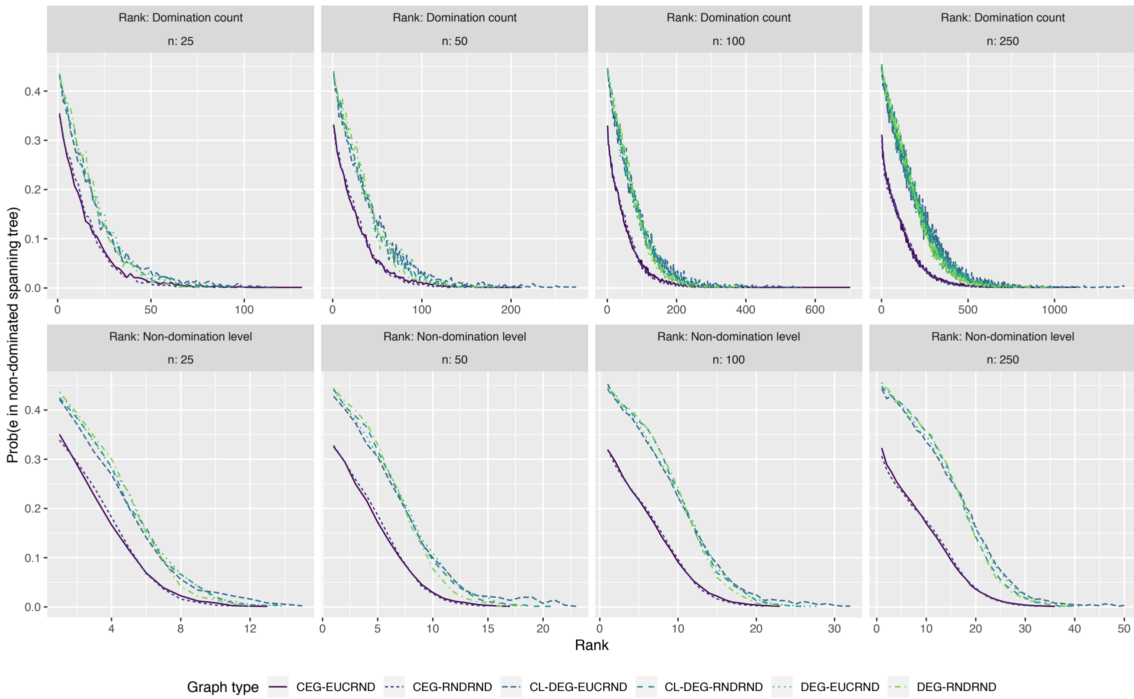

Figure 3: Empirical probability that an edge is part of at least one (supported) efficient spanning tree as a function of its non-domination level or domination number respectively (zero probabilities were filtered to smooth the estimates). The plots show probability estimations for different problem instance types and instance sizes n.

We estimated the probability of an edge to be part of an efficient spanning tree as a function of its rank. More precisely, for a single instance G and edge $e \in G$ we first calculated the share $s(e)$ of non-dominated spanning trees it is part of (see Eq. (1)). The probability of a rank-i edge is the average of shares of all edges of the corresponding rank. We repeated the above estimation for each 1000 randomly generated graphs of each instance group and instances size $n \in \{25, 50, 100, 250\}$. Hence, the probability of a rank-i edge for a specific instance type and size is the mean probability of the corresponding rank over all 1000 instances. A few words on Pareto-set computation: For instances with up to 50 nodes we calculated the Pareto-set with the exact multi-criteria Prim algorithm (see [12, p. 238]). However, for larger instances this approach became infeasible due to computational limitations. Hence, for $n > 50$ we relied on approximations of the Pareto-set instead of the true Pareto-set. Here, we adopted a weighted-sum approach with equidistant weights $\lambda_1 = \frac{k}{1000}, k = 0, 1, \ldots 1000$ and minimized $\lambda_1 c_1(T) + (1 - \lambda_1) c_2(T)$ with a single-objective algorithm. In a preprocessing step we filtered out duplicates within the approximation sets of 1001 supported efficient trees, since different weights may result in the same resulting non-dominated spanning tree.

Figure 3 depicts the empirical probabilities. For clarity only parts with probability greater $\varepsilon = 0.01$ are depicted for smoothing the curves. Otherwise, in case of the domination-count based ranking, the curves show strong oscillation. This can be explained as follows: non-domination levels are continuous in the sense that if rank i exists, ranks $1 \leq j < i$ necessarily exist. This property does not hold for the domination-count based ranking. The empirical probabilities present a clear picture. We observe that low-rank edges (for both non-domination level and domination count based ranks) have a much higher probability of occurrence in non-dominated spanning trees. However, in contrast to the single-objective case where a rank-1 edge is in a MST with probability 1 this is not the case in the multi-objective case due to the nature of trade-off solutions. Furthermore, Figure 3 reveals that for dense graphs (CEG) – regardless of the edge weight type (RNDRND or EUCRND) – the curve runs below the one for all remaining, less dense, graphs. This can be explained by the larger number of non-dominated spanning trees for CEG-graphs. This observation is supported by the exemplary scatterplots of edge weights in Figure 2 (bottom row) where for DEG-type graphs edges of higher rank are far more frequent members of non-dominated solutions. A close-up look at the curves reveals a subtle difference between RNDRND and EUCRND edge weighted graphs. While the probabilities for low-rank edges are higher for the former there is a position-swap for higher ranks where probabilities for the latter are higher (again this observation is independent of the adopted definition of rank). This effect is slightly stronger for clustered instances (CL-*) and can be explained as follows: There are only few edges interlinking clusters and these edges have a large first weight in EUCRND-type instances due to the distance of clusters. By chance the second objective is large, too. Since a spanning tree is connected, a fraction of those heavy edges is often a necessary part of non-dominated spanning trees. In summary, our observations support our intuition and assumptions. This motivates the design of biased edge-selection strategies and incorporation into the one-exchange operator in the next section.

4 BIASED EDGE SELECTION STRATEGIES

In this section we propose four biased edge-selection strategies for the one-edge-exchange mutation operator (see Section 2). Based on the insights gained by observing the space of Pareto-optimal spanning trees and the conducted rank probability estimations we implement a bias towards edges of low rank.

Rank Estimate (RE) Here we incorporate the empirical rank probabilities directly. Let p_r be the probability that an edge of non-domination level r is part of at least one efficient tree and $E_r \subset E$ be the set of edges located on non-domination level r. With this the probability of selection edge $e \in E$ for inclusion into the current candidate solution T is given by

$$p(e) = \frac{p_r + \varepsilon}{\sum_{\text{Rank } r} |E^r|(p_r + \varepsilon)}.$$

Note that we add a small constant $\varepsilon > 0$ to account for the possibility that certain ranks may not have occurred during the rank probability estimation due to randomness of instance generation.

Domination-Count Estimate (DE) This strategy is identical to RE but p_r is based on the domination-count based ranking.

Rank Simple (RS) The edge selection probability of edge $e \in E$ is proportional to the non-domination level of e, i. e.,

$$p(e) = \frac{l_{\max} - r(e) + 1}{\sum_{e \in E}(l_{\max} - r(e) + 1)}, \tag{2}$$

where $l_{\max}$ is the maximum non-domination level. Note that the linear transformation is performed because we prefer low-ranked edges. Note further, that the addition of 1 is necessary to allow the selection of edges with highest non-domination level.

Domination-Count Simple (DS) Identical to the RS-strategy (see Eq. 2). However, instead of non-domination level based ranking we use the domination-count based ranking.

5 EVALUATION

In order to evaluate the performance of all four proposed biased edge-selection strategies (DE, DS, RE and RS) in comparison to the uniform selection (UN) as baseline we performed a series of experiments.

5.1 Experimental Setup

We generated 25 instances of each combination of instance type and instance size considered in the rank estimation experiment series conducted in Section 3 ($6 \times 25 = 150$ instances in total). The well-known NSGA-II [10] served as a wrapper for the one-edge-exchange mutation operator[3], which was run with each edge-selection strategy 25 times on each problem instance. It should be noted that the i-th run was started with the same initial population for each mutation operator. The remainder of the setup is

[3]Recombination was omitted due to the potential bias of the results, since we focus on mutation exclusively.

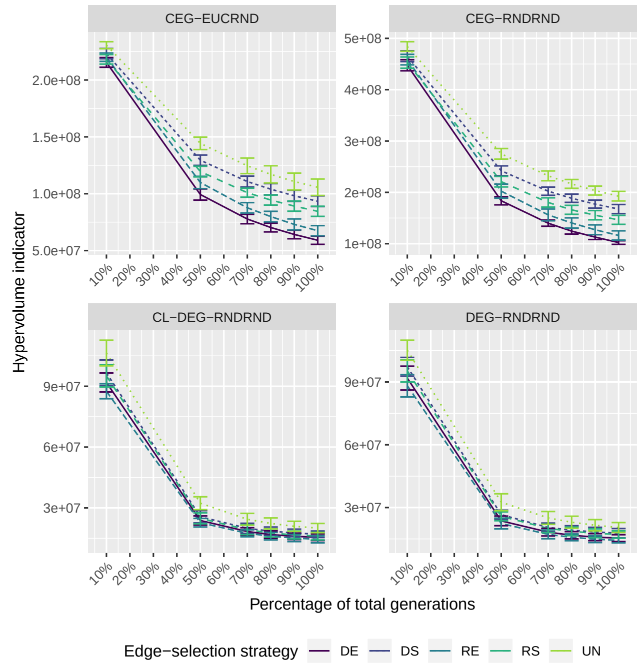

Figure 4: Exemplary optimization traces for each one instance of $n = 250$ nodes. We show the mean Hypervolume indicator values and 95% confidence bounds.

$\mu = 100, \lambda = 50$, mutation probability $p_m = 1$ and $100 \cdot n$ function evaluations served as the only stopping criterion. At discrete steps in time, 10%, 50%, 70%, 80%, 90% and 100% of total generations, we measured the progress of the EMOA by means of the Pareto-compliant hypervolume indicator (HV) [20] and the ε-indicator [20]. Solutions of 1000 runs of a weighted-sum scalarization approach with equidistantly chosen weights $\lambda_1 \in [0, 1]$ and $\lambda_2 = 1 - \lambda_1$ served as the reference set for the indicator computation.

The experiments were carried out with the software package `mcMST` [2]. The software `ecr` [1, 4] served as a toolbox for performance assessment of the experimental results.

5.2 Results

In the following we take several perspectives and report the results here. In a first approach, we consider the development of solution quality over time for all instances on all topologies. Figure 4 shows exemplary traces for the hypervolume indicator (HV) [20] measured over the whole runtime budget for the encapsulating EMOA. We show the mean traces over 25 runs for a representative instance out of each instance group with $n = 250$ nodes. For this instance size the traces show that all biased strategies outperform the baseline approach. This holds especially true for the observed instances that are strongly interconnected and dense (see CEG instances). Here, the domination-count estimate strategy (DE) is superior. Especially for clustered and the less dense DEG instances, differences in performance of the biased strategies are not always clearly identifiable in the comparison of averaged traces. Therefore, in a second step, we approach the analysis from a different perspective. We

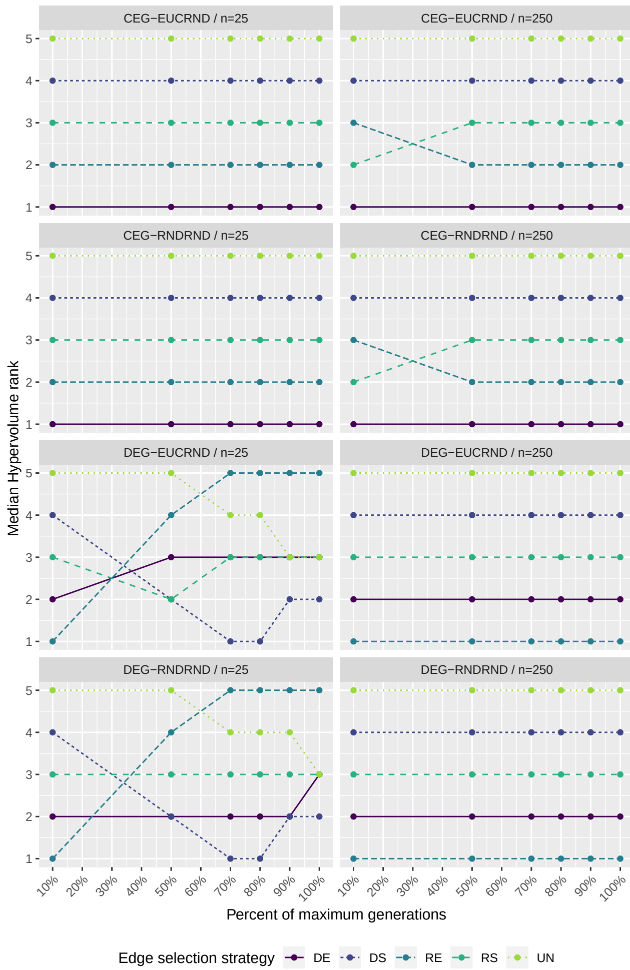

Figure 5: Parallel coordinate plots of the average of median Hypervolume ranks (lower is better) split up by instance group (rows) and instance size (columns).

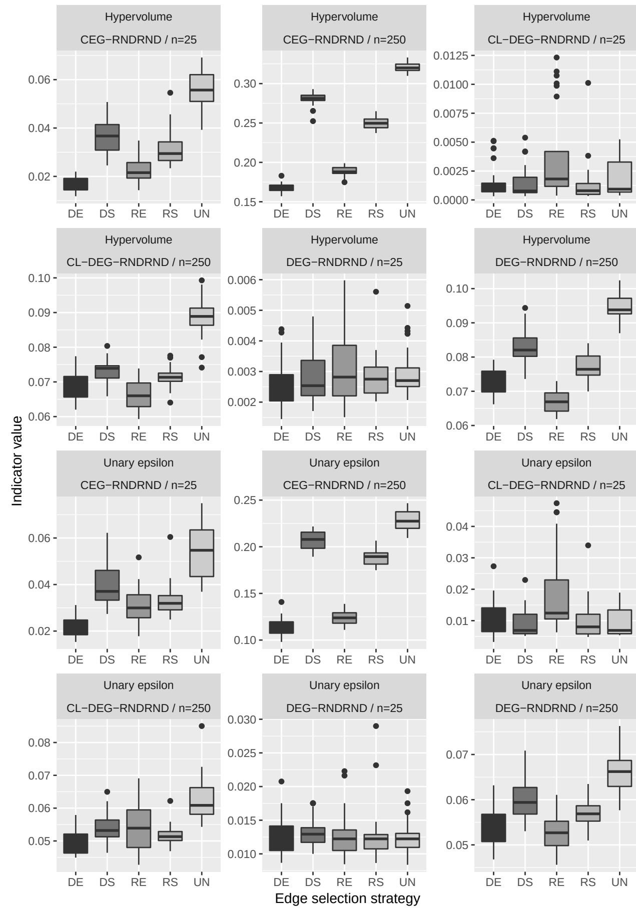

Figure 6: Distributions of hypervolume-indicator (top rows) and unary ε-indicator (bottom rows) for representative instances (the lower the better). Reference sets are based on the union of all approximation sets over all runs and edge-selections strategies.

rank the edge-selection strategies regarding mean hypervolume performance for each instance type and size and report the median hypervolume ranks per group and point in time (aggregation over all instances). The results shown in the parallel coordinate plots in Figure 5 confirm the observations for strongly connected and dense instances observed in Figure 4 for both large and small instances . However, for CL and DEG instances we can observe a superiority of the Rank Estimate (RE) on large instances. Interestingly, this effect does not hold for smaller instances. Here, the biased selection strategies often switch rank positions over time and specifically the winner for large instances (RE) performs worst of the four proposed approaches for small instances. The observation of the performance compared to uniform selection (UN) becomes most interesting for small instances ($n = 25$). While UN is outperformed at the beginning, its application seems to become beneficial later on. Figure 6 additionally validates the observation for the final results of EMOA runs. While in CEG instances all biased mutations outperform the uniform selection for hypervolume as well as the unary ϵ-indicator, UN is comparable to the biased operators for small and less connected/dense instances. For larger instances, UN is also outperformed on CL and DEG instances.

These findings are further detailed in Tables 2 and 3. In the first case, we again investigate the performance development over time and focus in Table 2 on three points in time during EMOA runs. At these points in time, we register the best performing edge-selection strategy with respect to hypervolume for the same initial population in each replication. This has been performed for all instance sizes $n \in \{25, 50, 100, 250\}$ and all instance groups. In Table 2, we show the percentage of best performance per condition. We find that in most cases, DE wins the competition against all other selection operators for connected/dense instances during the evolutionary process. But also for other instances and all instance sizes, the application of biased selection is overall helpful. Only in a minority of cases (and inversely related to instance size) UN is the overall superior selection operator.

Table 1: Median ranks of binary Hypervolume indicator (lower is better) for each combination of instance type and size and edge selection probability used after 50%/80%/100% of the number of generations.

	CEG-EUCRND				CEG-RNDRND				CL-DEG-EUCRND				CL-DEG-RNDRND				DEG-EUCRND				DEG-RNDRND			
Strategy	n = 25	n = 50	n = 100	n = 250	n = 25	n = 50	n = 100	n = 250	n = 25	n = 50	n = 100	n = 250	n = 25	n = 50	n = 100	n = 250	n = 25	n = 50	n = 100	n = 250	n = 25	n = 50	n = 100	n = 250
DE	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 1	1 / 1 / 2	2 / 3 / 3	1 / 2 / 2	2 / 2 / 3	1 / 1 / 1	2 / 2 / 3	2 / 2 / 2	2 / 2 / 2	1 / 1 / 1	3 / 3 / 3	2 / 2 / 2	2 / 2 / 2	1 / 1 / 1	2 / 2 / 3	2 / 2 / 2	1 / 1 / 1
DS	4 / 4 / 4	4 / 4 / 4	4 / 4 / 4	4 / 4 / 4	4 / 4 / 4	4 / 4 / 4	4 / 4 / 4	4 / 4 / 4	3 / 2 / 3	2 / 2 / 2	4 / 4 / 4	2 / 2 / 2	4 / 3 / 3	2 / 2 / 2	4 / 4 / 4	3 / 2 / 1	3 / 3 / 3	2 / 2 / 2	4 / 4 / 4	2 / 2 / 2	4 / 3 / 3	2 / 2 / 2	4 / 4 / 4	3 / 2 / 2
RE	2 / 2 / 2	2 / 2 / 2	2 / 2 / 2	2 / 2 / 2	2 / 2 / 2	2 / 2 / 2	2 / 2 / 2	2 / 2 / 2	4 / 4 / 4	4 / 5 / 5	2 / 1 / 1	4 / 4 / 4	2 / 3 / 4	3 / 4 / 4	1 / 1 / 1	4 / 4 / 4	4 / 4 / 4	4 / 5 / 5	1 / 1 / 1	4 / 4 / 4	2 / 4 / 4	4 / 5 / 5	1 / 1 / 1	4 / 4 / 4
RS	3 / 3 / 3	3 / 3 / 3	3 / 3 / 3	3 / 3 / 3	3 / 3 / 3	3 / 3 / 3	3 / 3 / 3	3 / 3 / 3	2 / 2 / 2	2 / 2 / 3	3 / 3 / 3	2 / 2 / 2	3 / 3 / 3	2 / 2 / 2	3 / 3 / 3	3 / 2 / 3	2 / 2 / 2	2 / 3 / 3	3 / 3 / 3	2 / 2 / 2	3 / 2 / 2	3 / 3 / 3	3 / 3 / 3	2 / 3 / 3
UN	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 4 / 4	5 / 5 / 5	5 / 4 / 4	5 / 5 / 5	5 / 5 / 4	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 3 / 3	5 / 5 / 5	5 / 5 / 5	5 / 5 / 5	5 / 4 / 3	5 / 5 / 5	5 / 5 / 5

Table 2: Percentage of runs where the corresponding edge selection strategies performed best regarding the hypervolume indicator split up by instance set and instance size. We report the results for 50%, 80% and 100% of the number of generations. Note that in each replication each edge selection strategy started with the same initial population. The highest percentage value is typeset bold with gray background for improved visual perception.

		50% generations					80% generations					100% generations				
group	n	RS	RE	DE	DS	UN	RS	RE	DE	DS	UN	RS	RE	DE	DS	UN
CEG-EUCRND	25	0.16%	30.40%	**69.44%**	0.00%	0.00%	0.32%	27.52%	**72.16%**	0.00%	0.00%	0.00%	25.12%	**74.88%**	0.00%	0.00%
	50	0.00%	13.92%	**86.08%**	0.00%	0.00%	0.00%	11.84%	**88.16%**	0.00%	0.00%	0.00%	9.92%	**90.08%**	0.00%	0.00%
	100	0.00%	2.88%	**97.12%**	0.00%	0.00%	0.00%	2.40%	**97.60%**	0.00%	0.00%	0.00%	1.60%	**98.40%**	0.00%	0.00%
	250	0.00%	0.80%	**99.20%**	0.00%	0.00%	0.00%	0.00%	**100.00%**	0.00%	0.00%	0.00%	0.16%	**99.84%**	0.00%	0.00%
CEG-RNDRND	25	0.16%	32.32%	**67.52%**	0.00%	0.00%	0.16%	25.92%	**73.92%**	0.00%	0.00%	0.00%	24.96%	**75.04%**	0.00%	0.00%
	50	0.00%	21.60%	**78.40%**	0.00%	0.00%	0.00%	15.20%	**84.80%**	0.00%	0.00%	0.00%	12.48%	**87.52%**	0.00%	0.00%
	100	0.00%	3.52%	**96.48%**	0.00%	0.00%	0.00%	3.04%	**96.96%**	0.00%	0.00%	0.00%	3.20%	**96.80%**	0.00%	0.00%
	250	0.00%	0.32%	**99.68%**	0.00%	0.00%	0.00%	0.32%	**99.68%**	0.00%	0.00%	0.00%	0.00%	**100.00%**	0.00%	0.00%
CL-DEG-EUCRND	25	22.08%	17.44%	**27.04%**	23.20%	10.24%	20.48%	16.00%	**24.64%**	23.84%	15.04%	22.08%	16.16%	21.76%	**23.20%**	16.80%
	50	**25.76%**	16.00%	24.96%	25.60%	7.68%	**27.84%**	14.08%	25.12%	22.24%	10.72%	**26.88%**	16.64%	21.76%	19.52%	15.20%
	100	25.12%	19.68%	**29.92%**	23.36%	1.92%	**28.96%**	14.40%	27.52%	24.80%	4.32%	26.88%	14.72%	**28.00%**	23.68%	6.72%
	250	22.08%	30.56%	**32.48%**	14.72%	0.16%	20.64%	**38.88%**	26.40%	12.32%	1.76%	19.20%	**46.56%**	23.52%	9.44%	1.28%
CL-DEG-RNDRND	25	25.28%	20.96%	**26.08%**	22.24%	5.44%	23.68%	14.24%	**29.28%**	25.12%	7.68%	24.16%	14.72%	22.72%	**26.72%**	11.68%
	50	22.24%	17.76%	**34.40%**	24.32%	1.28%	24.80%	12.64%	**30.72%**	29.12%	2.72%	26.56%	11.84%	**29.44%**	26.88%	5.28%
	100	18.56%	26.88%	**40.16%**	14.40%	0.00%	24.00%	18.56%	**39.36%**	17.76%	0.32%	26.24%	15.36%	**37.92%**	20.00%	0.48%
	250	8.16%	**55.84%**	33.92%	2.08%	0.00%	6.08%	**64.48%**	26.88%	2.56%	0.00%	5.60%	**74.08%**	19.20%	1.12%	0.00%
DEG-EUCRND	25	22.24%	15.20%	22.08%	**27.68%**	12.80%	20.96%	11.04%	20.96%	**27.84%**	19.20%	23.04%	8.96%	21.92%	**26.08%**	20.00%
	50	24.80%	16.16%	**32.48%**	23.20%	3.36%	26.72%	13.60%	**30.08%**	23.20%	6.40%	24.64%	12.96%	**28.96%**	25.76%	7.68%
	100	24.48%	15.68%	**36.48%**	21.76%	1.60%	28.48%	14.56%	**30.08%**	24.00%	2.88%	26.24%	18.72%	**29.28%**	21.92%	3.84%
	250	13.76%	**44.64%**	37.60%	4.00%	0.00%	10.72%	**59.52%**	26.72%	3.04%	0.00%	7.36%	**70.40%**	19.84%	2.40%	0.00%
DEG-RNDRND	25	20.64%	15.52%	**28.96%**	26.88%	8.00%	21.28%	10.08%	**27.20%**	27.04%	14.40%	20.48%	10.40%	24.32%	**28.32%**	16.48%
	50	22.88%	17.12%	**36.96%**	20.96%	2.08%	21.92%	11.84%	**36.96%**	25.76%	3.52%	21.92%	10.24%	**35.84%**	26.88%	5.12%
	100	20.32%	22.40%	**43.04%**	14.24%	0.00%	22.56%	16.00%	**43.68%**	17.76%	0.00%	24.96%	16.32%	**41.12%**	17.44%	0.16%
	250	8.16%	**51.52%**	38.88%	1.44%	0.00%	7.20%	**60.32%**	31.04%	1.44%	0.00%	6.72%	**70.24%**	22.24%	0.80%	0.00%

Table 3 details the pairwise comparison of performance of the selection operators regarding the HV indicator. For testing, we use the non-parametric Wilcoxon rank sum test testing the hypothesis pair $H_0 : I_{\text{HV}}(A, I) \geq I_{\text{HV}}(B, I)$ vs. $H_1 : I_{\text{HV}}(A, I) < I_{\text{HV}}(B, I)$ for edge-selection strategies A, B and instance I with significance level $\alpha = 0.01$ (with Bonferroni-Holm p-value adjustment). The percentage values in Table 3 aggregate for the given pair of biased selection mechanisms, the instance type, and the respective instance size, at what ratio the first selection operator significantly outperforms the second operator, i. e., the fraction of instances for which the null hypothesis H_0 was rejected. The unbiased UN operator is almost always significantly outperformed and does only once, in a single run for a small instance of $n = 25$ with low density significantly outperform all other strategies. The results show that the DE selection is constantly superior to all other biased selection types for complete/dense graphs. Additionally the DE and DS selection are slightly more stable in its quality compared to baseline (UN).

Overall, from the analysis and for mcMST problems, we can conclude that biased selection strategies for edges included into the one-edge-exchange mutation are advantageous and support convergence to the optimal solution. For larger and strongly connected/dense instances this effect is more pronounced than for small and sparse graph topologies. Still, all biased selection approaches significantly outperform the standard approach of unbiased mutation in the vast majority of cases[4]. This essentially means, that beneficial information – that is, the sampling probability for edges – can be extracted from the considered problem instance and directly integrated into the acceleration of the evolutionary solution approach.

6 CONCLUSIONS

We considered biased edge-selection strategies for the classical one-edge-exchange mutation operator for the bi-criteria spanning tree problem. Experiments show that – similar to the single-objective scenario as research indicated in earlier studies – edges of "low

[4] As shown in Table 3, there were only few special cases in which UN was not significantly outperformed.

Table 3: Aggregated results of pairwise non-parametric Wilcoxon rank sum tests with Bonferroni-Holm p-value adjustment to account for potential multiple testing issues. The table shows the fraction of instances I (split by instance set and instance size) where the corresponding null hypothesis $I_{\text{HV}}(A, I) \geq I_{\text{HV}}(B, I)$ is rejected as highly significant ($\alpha = 0.01$). For better visual perception, values $> 20\%$ are highlighted with increasing background intensity.

group	n	DE vs. DS	DS vs. DE	DE vs. RE	RE vs. DE	DE vs. RS	RS vs. DE	DE vs. UN	UN vs. DE	DS vs. RE	RE vs. DS	DS vs. RS	RS vs. DS	DS vs. UN	UN vs. DS	RE vs. RS	RS vs. RE	RE vs. UN	UN vs. RE	RS vs. UN	UN vs. RS
CEG-EUCRND	25	**100%**	0%	60%	0%	**100%**	0%	**100%**	0%	0%	**100%**	0%	76%	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
	50	**100%**	0%	**92%**	0%	**100%**	0%	**100%**	0%	0%	**100%**	0%	**100%**	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
	100	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%	0%	**100%**	0%	**100%**	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
	250	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%	0%	**100%**	0%	**100%**	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
CEG-RNDRND	25	**100%**	0%	64%	0%	**100%**	0%	**100%**	0%	0%	**100%**	0%	88%	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
	50	**100%**	0%	**96%**	0%	**100%**	0%	**100%**	0%	0%	**100%**	0%	**100%**	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
	100	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%	0%	**100%**	0%	**100%**	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
	250	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%	0%	**100%**	0%	**100%**	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
CL-DEG-EUCRND	25	4%	4%	16%	0%	0%	4%	4%	0%	12%	0%	0%	0%	16%	4%	0%	16%	0%	16%	8%	0%
	50	0%	0%	8%	0%	0%	0%	4%	0%	20%	0%	0%	4%	16%	0%	0%	20%	0%	12%	16%	0%
	100	0%	0%	20%	0%	4%	0%	52%	0%	16%	0%	4%	0%	44%	0%	0%	16%	24%	4%	48%	0%
	250	16%	0%	0%	24%	4%	0%	**100%**	0%	0%	68%	0%	8%	72%	0%	40%	0%	**100%**	0%	**92%**	0%
CL-DEG-RNDRND	25	0%	4%	8%	0%	8%	0%	28%	0%	28%	0%	4%	0%	36%	0%	0%	16%	12%	4%	20%	0%
	50	0%	0%	44%	0%	0%	0%	80%	0%	44%	0%	0%	0%	72%	0%	0%	36%	24%	0%	56%	0%
	100	20%	0%	40%	0%	4%	0%	**100%**	0%	24%	0%	0%	4%	**100%**	0%	0%	32%	80%	0%	**100%**	0%
	250	**100%**	0%	0%	68%	60%	0%	**100%**	0%	0%	**100%**	0%	68%	**100%**	0%	**100%**	0%	**100%**	0%	**100%**	0%
DEG-EUCRND	25	0%	4%	12%	0%	0%	0%	0%	4%	28%	0%	4%	0%	4%	4%	0%	12%	0%	24%	0%	4%
	50	0%	0%	12%	0%	0%	0%	36%	0%	20%	0%	4%	0%	60%	0%	0%	20%	8%	4%	40%	0%
	100	0%	0%	28%	0%	4%	0%	84%	0%	24%	0%	0%	0%	76%	0%	0%	24%	56%	8%	84%	0%
	250	88%	0%	0%	68%	16%	0%	**100%**	0%	0%	**100%**	0%	24%	**100%**	0%	**96%**	0%	**100%**	0%	**100%**	0%
DEG-RNDRND	25	0%	0%	36%	0%	0%	0%	0%	0%	44%	0%	4%	0%	8%	0%	0%	32%	0%	40%	4%	4%
	50	8%	0%	64%	0%	4%	0%	84%	0%	56%	0%	0%	0%	88%	0%	0%	44%	24%	0%	80%	0%
	100	24%	0%	44%	0%	8%	0%	**100%**	0%	20%	0%	0%	4%	**100%**	0%	0%	32%	80%	0%	**100%**	0%
	250	**100%**	0%	0%	68%	64%	0%	**100%**	0%	0%	**100%**	0%	60%	**100%**	0%	88%	0%	**100%**	0%	**100%**	0%

rank", i. e., a low non-domination level or a low number of dominating edges according to our definition, have a much higher probability of occurrence in non-dominated spanning trees. We leveraged these insights by proposing several biased edge-selection strategies and compared their performance with the unbiased uniform edge-selection strategy. A study on a diverse set of benchmark graphs reveals a significant superiority over the baseline in particular for complete graphs with respect to hypervolume and ε -indicator in particular for graphs with high edge density.

Future research directions are manifold: In this work we focused on the edge-selection probability for insertion. However, removing an edge from the resulting cycle was still based on a uniform random decision. There are several avenues for improvement in this direction, e. g., by mirroring the edge-selection probability for insertion or deleting edges in some greedy fashion. Moreover, we are excited to investigate whether our approach is directly transferable to other, more complex, combinatorial optimization problem like the multi-criteria Traveling Salesperson Problem.

REFERENCES

[1] Jakob Bossek. 2017. Ecr 2.0: A Modular Framework for Evolutionary Computation in R. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) Companion*. ACM, Berlin, Germany, 1187–1193. https://doi.org/10.1145/3067695.3082470

[2] Jakob Bossek. 2017. mcMST: A Toolbox for the Multi-Criteria Minimum Spanning Tree Problem. *The Journal of Open Source Software* 2, 17 (2017), 1–2. https://doi.org/10.21105/joss.00374

[3] Jakob Bossek. 2018. grapherator: A Modular Multi-Step Graph Generator. *The Journal of Open Source Software* 3, 22 (2018), 528. https://doi.org/10.21105/joss.00528

[4] Jakob Bossek. 2018. Performance Assessment of Multi-objective Evolutionary Algorithms with the R Package Ecr. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '18)*. ACM, Kyoto, Japan, 1350–1356. https://doi.org/10.1145/3205651.3208312

[5] Jakob Bossek and Christian Grimme. 2017. A Pareto-Beneficial Sub-Tree Mutation for the Multi-Criteria Minimum Spanning Tree Problem. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, Honolulu, HI, USA, 3280–3287. https://doi.org/10.1109/SSCI.2017.8285183

[6] Pedro Cardoso, Mário Jesus, and Álberto Márquez. 2006. *Multiple criteria minimum spanning trees.* Technical Report 2006-01. University of Algarve, Computing and Simulation Center CsC.

[7] A. Cayley. 1889. A theorem on trees. *Quart. J. Math* 23 (1889), 376âĂŞ378.

[8] 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

[9] Kalyanmoy Deb. 2012. *Optimization for Engineering Design - Algorithms and Examples, Second Edition.* PHI Learning Private Limited.

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[11] Boris N. Delaunay. 1934. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR* 6 (1934), 793–800.

[12] Matthias Ehrgott. 2006. *Multicriteria Optimization.* Vol. 5. 329 pages. https://doi.org/10.1007/3-540-27659-9 arXiv:1105.4109

[13] Hisao Ishibuchi, Yasuhiro Hitotsuyanagi, Noritaka Tsukamoto, and Yusuke Nojima. 2009. Use of Biased Neighborhood Structures in Multiobjective Memetic Algorithms. *Soft Comput.* 13, 8-9 (March 2009), 795–810. https://doi.org/10.1007/s00500-008-0352-6

[14] J. D. Knowles and D. W. Corne. 2001. A comparison of encodings and algorithms for multiobjective minimum spanning tree problems. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, Vol. 1. 544–551 vol. 1. https://doi.org/10.1109/CEC.2001.934439

[15] 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.

[16] 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.

[17] Stefan Ruzika and Horst W. Hamacher. 2009. Algorithmics of Large and Complex Networks. Springer-Verlag, Berlin, Heidelberg, Chapter A Survey on Multiple Objective Minimum Spanning Tree Problems, 104–116. https://doi.org/10.1007/978-3-642-02094-0_6

[18] Michael Stein. 1987. Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics* 29, 2 (1987), 143–151.

[19] Gengui Zhou and Mitsuo Gen. 1999. Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research* 114, 1 (1999), 141 – 152. https://doi.org/10.1016/S0377-2217(98)00016-2

[20] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7, 2 (2003), 117–132.