A Pareto-Beneficial Sub-Tree Mutation for the Multi-Criteria Minimum Spanning Tree Problem

Jakob Bossek and Christian Grimme
University of Münster, Department of Information Systems
48149 Münster, Germany
Email: {bossek, christian.grimme}@uni-muenster.de

***Abstract*—While finding minimum-cost spanning trees (MST) in undirected graphs is solvable in polynomial time, the multi-criteria minimum spanning tree problem (mcMST) is NP-hard. Interestingly, the mcMST problem has not been in focus of evolutionary computation research for a long period of time, although, its relevance for real world problems is easy to see. The available and most notable approaches by Zhou and Gen as well as by Knowles and Corne concentrate on solution encoding and on fairly dated selection mechanisms. In this work, we revisit the mcMST and focus on the mutation operators as exploratory components of evolutionary algorithms neglected so far. We investigate optimal solution characteristics to discuss current mutation strategies, identify shortcomings of these operators, and propose a sub-tree based operator which offers what we term Pareto-beneficial behavior: ensuring convergence and diversity at the same time. The operator is empirically evaluated inside modern standard evolutionary meta-heuristics for multi-criteria optimization and compared to hitherto applied mutation operators in the context of mcMST.**

I. Introduction

Finding the minimal spanning tree (MST) of a weighted graph is a basic problem of operations research. Given an edge-weighted graph $G = (V, E, c)$ with cost function $c : E \to \mathbb{R}^m$, the problem is to determine a cost-efficient tree connecting all vertices from V by using a subset of $|V| - 1$ edges from E. The relevance of solutions for this problem is obvious for network design problems, e. g. in the context of planning electric power grids or wiring data and communication networks. If only a single cost factor is considered per edge ($m = 1$), the problem is solvable in polynomial time. Very popular approaches by Prim [1] and Kruskal [2] perform greedy searches on the edge list of the given graph. They select feasible[1] and most cost-efficient edges to be added to the spanning tree.

For real-world applications in network design, more than a single objective has to be considered. For an electrical power grid the length of edges (i. e. the material costs) are certainly important. However, also environmental costs, costs induced by regulatory laws, landscape properties, or other criteria have similar importance. Respecting all these criteria directly leads to the multi-criteria minimum spanning tree (mcMST) problem, in which a cost vector is given for each edge. Finding the Pareto-set for mcMST is NP-hard [3].

[1]I. e., edges which are not yet selected and do not close a cycle in the partial solution tree.

As Ruzika and Hamacher report in a comprehensive survey [4], a lot of attention has been paid to the multi-criteria problem by the operations research community. In contrast, Knowles and Corne [5] report (in their own work on mcMST from 2001) of only a single approach using evolutionary computation to tackle this problem. This first approach by Zhou and Gen [6] adopts a permutation encoding for trees proposed by Prüfer [7] and an early version of the non-dominated sorting genetic algorithm (NSGA) proposed by Srinivas and Deb [8]. Knowles and Corne themselves propose a second approach based on their own Pareto archived evolutionary strategy (PAES) [9] and drop the Prüfer encoding in favor of a direct encoding, i. e., a list of $|V| - 1$ edges contained in the spanning tree. They find, that direct encoding is more effective than the approach by Zhou and Gen. Consistent with these empirical findings, Gottlieb et al. [10] argue that Prüfer encoding is, in fact, inferior for spanning trees in evolutionary algorithms. They show, that Prüfer codes do not represent similar spanning trees under standard evolutionary variation operators.

Surprisingly, since Knowles' and Corne's work, new evolutionary approaches for handling the mcMST problem are rather scarce. Han and Wang [11] propose a so-called novel genetic algorithm (NGA), which is based on pure non-domination selection[2] and incorporates adapted crossover and local search mechanisms on a direct encoding of spanning trees. However, those mechanisms are not clearly motivated or methodologically founded. Mutation is done purely random. Chen et al. [12] revisit and (as they claim) improve the algorithm of Zhou and Gen [6]. The authors use the Prüfer encoding and propose a dislocation crossover, which interchanges only two arbitrary values of parental chromosomes for generating offspring. Cardoso et al. [13] report on an ant colony optimization (ACO) approach adopted for the multi-criteria case. Therein, the authors concentrate on the very ACO-specific design of their approach and compare it to a (single-objective) weighted sum approach.

Overall, most approaches concentrate on algorithmic design aspects like choosing an adequate meta-heuristic or finding a sophisticated selection mechanism. Mutation as a key ingredient of search and exploration is not investigated. However, it

[2]Note, that the authors do not integrate a diversity preservation mechanism. Thus, the approach is only comparable to very early multi-criteria approaches before NSGA.

978-1-5386-2726-6/17/$31.00 ©2017 IEEE

is well known, that problem tailored mutation operators and/or the integration of local search in (combinatorial) multi-criteria algorithms can significantly improve on solution quality and convergence towards the Pareto-front. E. g., Liu et al. [14] use highly specialized mutation operators to solve a multi-criteria version of the bin packing problem and Grimme et al. [15] apply single-objective priority rules in mutation operators for machine scheduling problems. In this work, we will close this gap for the mcMST by revisiting the evolutionary solution of the problem and specifically focus on mutation.

Therefore, we will have a look at true Pareto-optimal solutions for mcMST in order to learn about properties of Pareto-optimal solutions. In that light, we can discuss the seemingly state-of-the-art edge exchange mutation and motivate a Pareto-beneficial[3] mutation operator on direct encoding for the mcMST problem. This new operator is based on sub-tree mutation. For an evaluation independent of the chosen meta-heuristic, we adopt two modern evolutionary multi-criteria algorithms to integrate the available mutation operators: the sub-tree mutation proposed here, the standard edge exchange mutation, and Zhou and Gen's Prüfer-code-based mutation operator. For comparison, we use carefully generated benchmarks.

II. Multi-criteria minimum spanning trees

Let $G = (V, E, c)$ be an undirected, complete graph vertex set V, edge set E and vector-valued cost function $c : E \to \mathbb{R}^m$, $c(e) = (c_1(e), \ldots, c_m(e))$. Each acyclic, connected subgraph $T \subset G, T = (V, E_T)$ with $E_T \subset E$ is termed a *spanning tree* of G. In the following, we occasionally identify a tree by its edge set E_T. Further, $\mathcal{T}$ denotes the set of all spanning trees of G. With slight abuse of notation[4] the cost-vector of a tree $T \in \mathcal{T}$ contains the component-wise sum of edge weights of T, i. e., $c(T) = (c_1(T), \ldots, c_m(T))$ with $c_i(T) := \sum_{e \in E_T} c_i(e)$. Then the problem

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

is termed the *multi-criteria minimum spanning tree problem*. For $m = 1$, i. e., in the single-objective case, the solution is obvious. However, since there is no canonical order in $\mathbb{R}^m, m \geq 2$, we adopt the concept of Pareto-dominance to define optimality in the multi-criteria case. Here, the goal is to find a set of incomparable trade-off solutions $PS = \{T \in \mathcal{T} \,|\, \nexists\, T' \in \mathcal{T} : c(T') \preceq c(T)\}$, termed Pareto-set and its image $c(PS) = \{c(T) \,|\, T \in PF\}$. Here $\preceq$ is the *dominance relation*. A solution T dominates another solution T', $T \preceq T'$, if $c_i(T) \leq c_i(T') \, \forall i = 1, \ldots, m$ and $\exists j \in \{1, \ldots, m\}$ with $c_j(T) < c_j(T')$.

A. Prim's algorithm

The single-objective minimum spanning tree problem (MST) is solvable in polynomial time, e. g., using Prim's algorithm [1]. The algorithm maintains two disjoint sets of nodes C and $U = V \setminus C$ and an edge set E_T. Set C contains nodes already added to the spanning tree, while U holds the nodes not included, yet. Initially, C contains a single arbitrary node and $E_T = \emptyset$. Then, Prim's algorithm iteratively determines an edge, say $e^* = (v, u)$, that connects a node $v \in V$ with a node $u \in U$ favoring minimum weights $c(e^*) = \min_{e \in V \times U} c(e)$. Next, node u is inserted into C (and hence removed from U) and e^* is inserted into E_T. A minimum spanning tree $T := (C, E_T)$ of graph G is returned after $|V| - 1$ iterations.

B. Tree encodings and mutation

Cayley's theorem [16] states, that the set $\mathcal{T}$ of distinct spanning trees for a complete graph with nodes V has cardinality $|V|^{|V|-2}$. Prüfer [7] carried out a constructive proof of Cayley's theorem, which yields a bijection between a set of strings comprising $n - 2$ node labels from $\{1, \ldots, |V|\}$ and $\mathcal{T}$. This *Prüfer code* was adopted by Zhou and Gen as genotype representation for their evolutionary algorithm [6]. This representation benefits from its simplicity and allows for the use of standard mutation operators. Zhou and Gen use a simple mutation scheme, i. e., the operator replaces a randomly selected digit with a randomly sampled digit from $\{1, \ldots, |V|\}$ yielding another spanning tree. As reported by Gottlieb et al. [10], a major drawback of the Prüfer-representation is its poor locality. A small genotypic mutation may result in a huge phenotypic diversion.

Direct encoding, i. e., identifying a spanning tree with an edge list, is another straight-forward encoding used by Knowles and Corne [5] in their work on the mcMST. They used an edge-exchange mutation, which works as follows: a randomly selected edge is deleted from a given solution entailing a decomposition into two components. In order to reconnect those components, a random edge (excluding the before dropped edge) between both components is added.

III. Empirical properties of efficient mcMST solutions

Mutation operators in evolutionary algorithms can be considered as central drivers of innovation in populations. However, in multi-criteria optimization, the design of elaborated selection mechanisms is often in the main focus of research, while exploration in decision space is often neglected. Instead simple random variation strategies are employed. All too often, this gives away potential speed-up induced by smart mutation mechanisms.

As starting point for the investigation of mutation properties, we empirically analyze optimal solutions for the mcMST and strive to learn about genotypic properties of optimal solutions forming the Pareto-front. In a next step, we use these insights for creating a beneficial mutation operator fitted for mcMST. Note that we restrict our analysis on the direct encoding of spanning trees, as Gottlieb et al. [10] have shown that the Prüfer encoding is inferior for the representation of spanning trees in evolutionary algorithms.

[3] I. e., the mutation operator only generates dominating or incomparable solutions.

[4] The domain of c is actually the cartesian product V^2 and not a set of trees.

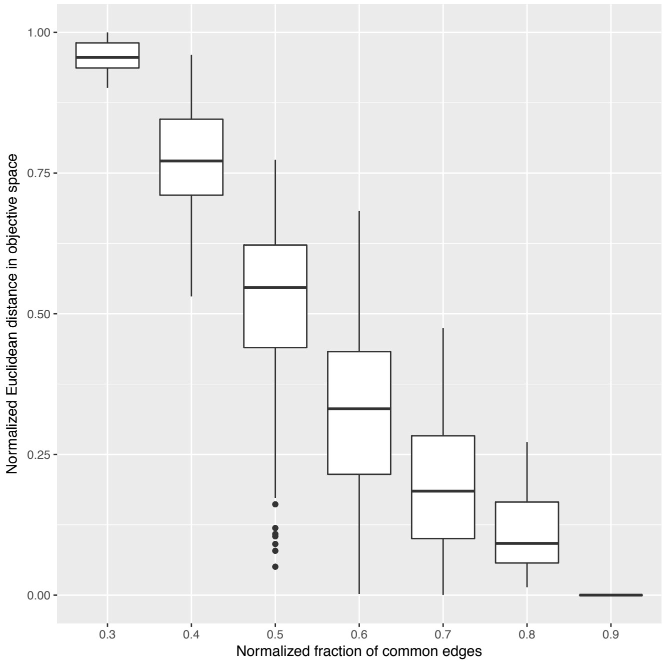


Fig. 1. Boxplots of the distribution of Euclidean distances between Pareto-optimal points in the objective space separated by the fraction of common edges. In case of few common edges the distances between solutions on the Pareto-front are quite high, while neighboring Pareto-optimal solutions in objective share many edges.

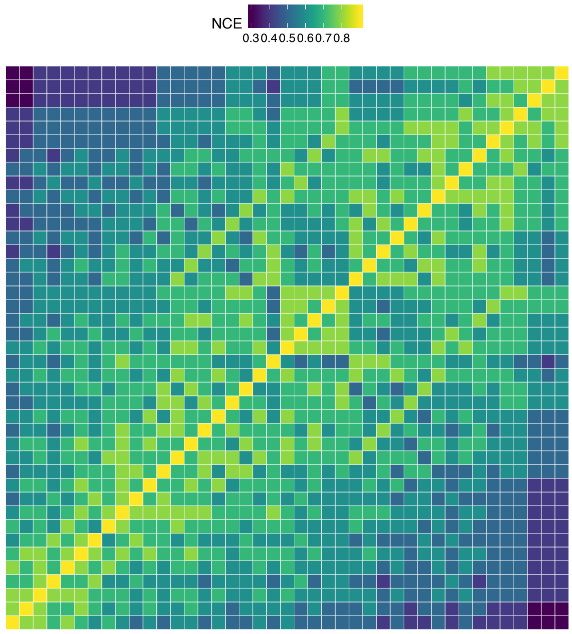


Fig. 2. Heatmap on the pairwise similarity between all Pareto-optimal solutions of an instance with $n = 10$ nodes. The solutions are displayed in ascending order regarding objective one and thus in descending order regarding objective two.

A. Problem instance generation

We generate random bi-criteria complete graphs following [6] and [5]. The first cost component, which contributes to the first objective function, is sampled uniformly randomly distributed as $c_1(u, v) \in \mathcal{U}(10, 100)$ while the second cost component is sampled as $c_2(u, v) \in \mathcal{U}(10, 50)$ for $(u, v) \in E$. Methods to generate the instance are available in the R-package **mcMST** [17] provided together with this work.

B. Analysis and conclusions for mutation

We empirically analyze properties of optimal solutions forming the Pareto-front based on ten bi-criteria mcMST instances, which comprise 10 nodes and are created with the before described procedure.

In order to compute the Pareto-front, a simple enumeration method was applied to each instance: using the Prüfer code representation, we generated all possible permutations (i. e., all possible spanning trees) and evaluated them with respect to non-domination. This yielded the optimal Pareto-front of multi-criteria spanning trees for these instances. Note that we used the Prüfer representation of spanning trees only for easy enumeration of all permutation. In all following steps, the spanning trees are considered in direct representation (i. e., as list of edges).

The comparison of optimal solutions on the Pareto-front starts with a definition of two simple measures. The *normalized distance of solutions* (ND) in objective space is computed as Euclidean distance between two solutions normalized with the Euclidean distance between the lexicographic optima of the Pareto-front. This measure only matters in our analysis when it is combined with the second measure: the *normalized common edges* (NCE). This measure of similarity counts the absolute number of common edges in two given spanning trees normalized by $|V|-1$ (the number of edges in a spanning tree).

The defined metrics are used to visualize the similarity of solutions based on their positions in the Pareto-front. Figure 1, shows the relationship of solution distance in objective space and solution similarity in decision space. Additionally, Figure 2 offers a more detailed view on similarity of neighboring solutions in the Pareto-front. For bi-criteria problems, neighborhood properties of solutions in the Pareto-front are preserved, when the solutions are simply ordered regarding one objective. Then, each box in Figure 2 represents the neighborhood relation between two solutions. Further, each box is colored with respect to the normalized common edges metric. This produces a heatmap view.

Both views provide first insights into the optimal solution construction:

1) The larger the distance of solutions in objective space is, the less similar spanning trees (which encode these solutions) are, see Figure 1.
2) At the same time, the detailed view provided in Figure 2 supports these findings on the level of individual solution comparison. The lexicographic optima of the Pareto-front are also most different (see lower right and upper left corner of the heatmap). However, even most contradicting trade-offs comprise common edges in the

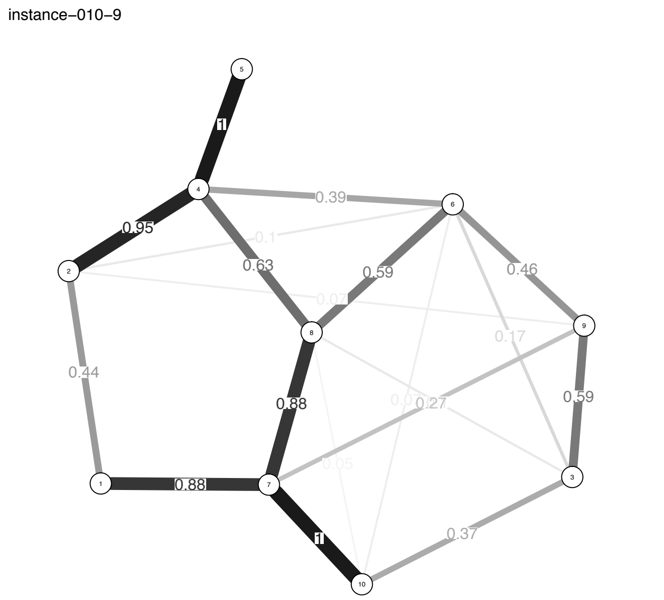


Fig. 3. Embedding of an examplary instance with ten nodes. The displayed edges are annotated with the fraction of Pareto-optimal solutions which contain the particular edges. Edges which are not part of any Pareto-optimal spanning tree are omitted here.

spanning tree.

These observations confirm the rather strong neighborhood relationship of solutions in decision space (with respect to common edges) and objective space (regarding their position in the Pareto-front). Thus, for a direct encoded spanning tree, small changes to the edge list may also lead to nearby solutions in objective space. Consequently, an edge exchange mutation operator seems to be a logical choice to perform small changes to solutions in an evolutionary heuristic.

However, a detailed analysis of the solution phenotypes reveals also disadvantages of the edge exchange mutation: Figure 3 shows the relative occurrence frequency for each edge over all spanning trees in the Pareto-front of an exemplary graph instance. Only edges shown in Figure 3 actually occur in Pareto-optimal spanning trees for the depicted instance. Each edge is annotated with the occurrence frequency over all solutions. Interestingly, many edges are often part of Pareto-optimal solutions. Some even seem to be constant members of all Pareto-optimal spanning trees while many are never part of any Pareto-optimal solution. This behavior is related to the costs vectors of the edges in a given graph instance. As shown in Figure 4 for all evaluated instances of size 10, the occurrence frequency often correlates with the contradictory character and domination level of cost vectors. Note that although the observations are based on very small instances, the insights are supported by the results in SectionV.

Reconsidering the working principles of the edge exchange operator, we can now identify several disadvantages:

1) Random inclusion and removal of edges from the spanning tree edge list may lead to a removal of constant or near constant edges. It can also include edges, which are never part of an optimal solution. This automatically leads to a deterioration of solution quality and hinders convergence.
2) The simple random character of edge exchange mutation slows down the inclusion of constant edges into solutions.

In order to address these issues for mutation systematically, we propose a new mutation operator in the next section.

IV. An mcMST sub-tree mutation operator

Based on the observations and insights gathered in the preceeding analysis of Pareto-optimal solutions and edge exchange mutation, we propose a sub-tree-based mutation operator. This section describes its working principles and properties in detail.

A. Technical details

In a nutshell, the proposed *sub-tree mutation* operator randomly selects a connected sub-tree of a solution and replaces it with the minimum-cost sub-tree regarding one objective.

Algorithm 1 mcMST sub-tree mutation

Require: Graph $G = (V, E, c = (c_1, \ldots, c_m))$, solution edge list E_T

1: $e_r :=$ select random edge from E_T
2: $V_s := \{v, w\}$ with $e_r = (v, w) \in E_T$
3: $\sigma \sim \mathcal{U}_{\mathcal{Z}}(3, \lfloor(|V| - 1)/2\rfloor)$
4: **while** $|V_s| < \sigma$ **do**
5: $\quad V_s = V_s \cup \{v \mid (v, w) \in E_T \text{ and } w \in V_s\}$
6: $o \sim \mathcal{U}_{\mathcal{Z}}(1, m)$ ▷ Sample random objective
7: $G' := (V_s, E_s = V_s \times V_s, c_o)$
8: $(V_s, E_s^*) = \text{PRIM}(G')$
9: **return** $(E_T \setminus E_s) \cup E_s^*$

Algorithm 1 presents a more detailed outline of the working principle. Given a problem instance $G = (V, E, c)$ and the edge list of a spanning tree of G, the operator selects a random edge $e_r = (v, w)$ from the edge list (line 1). Then, a set V_s is initialized with the incident nodes of e_r (line 2). V_s will be used throughout the algorithm as container for the nodes of the selected sub-tree. Next, a value σ is sampled, which represents a soft constraint for the size of the selected sub-tree. The main loop (lines 4-5) increases the selected sub-tree by extending V_s with nodes adjacent to nodes already contained in V_s. Adjacency is computed only w. r. t. edges contained in the considered spanning tree. The loop terminates, if at least σ nodes are contained in V_s. This approach ensures, that the selected sub-tree is connected. Following, a single objective is uniformly randomly sampled in line 7. For only the selected objective, Prim's algorithm[5] is applied to the nodes contained in V_s considering all connecting edges from the weighted

[5]Clearly, any alternative single-objective MST algorithm is applicable as well.

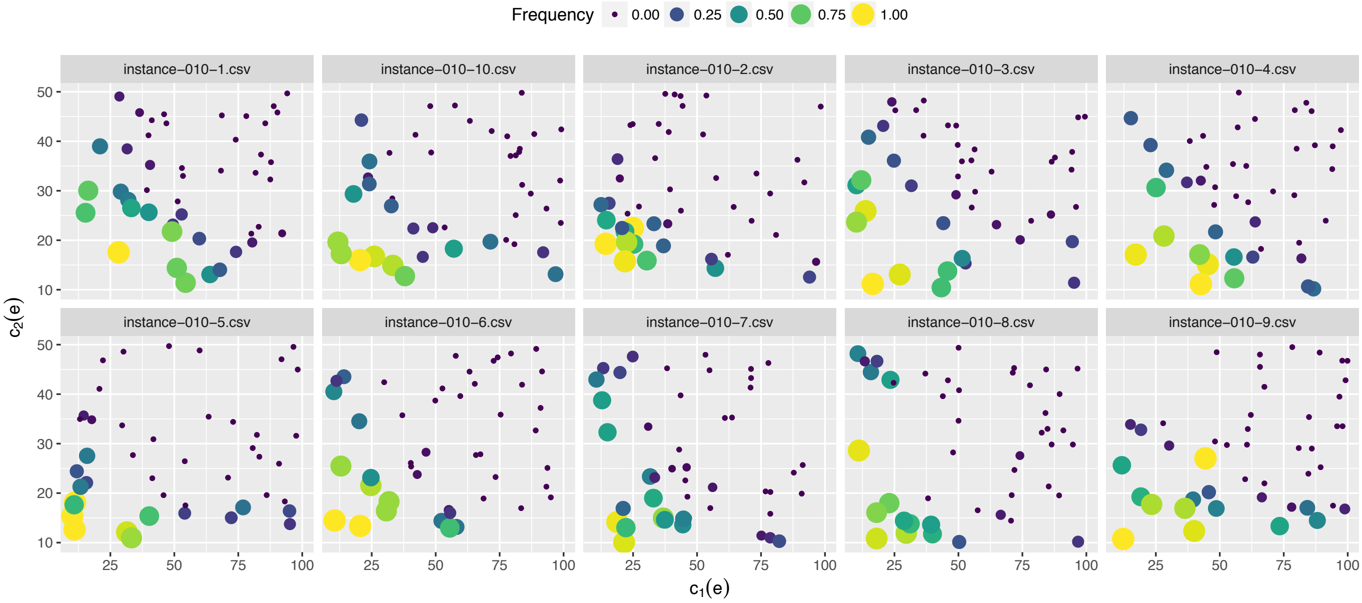
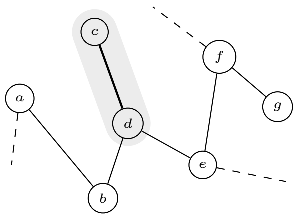
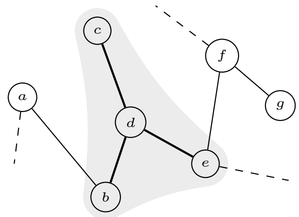


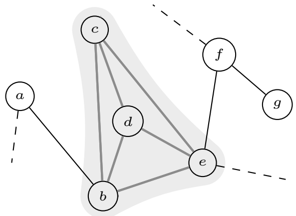
Fig. 4. Scatterplot of edge weights $c_i(e), i = 1, 2$ of instances with $|V| = 10$ nodes, i. e., each point represents one edge cost configuration. The points vary in size and color indicating the fraction of Pareto-optimal spanning trees in which the edge is contained.



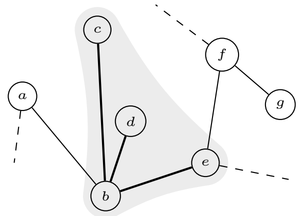
(a) Selection of initial edge e_r.



(b) Collecting a sub-tree with $|V_s| \geq \sigma$ to be replaced.



(c) Apply Prim's algorithm to all available edges connecting edges from $|V_s|$ for a single objective.



(d) Replace original sub-tree by the result from (c).

Fig. 5. Examplary application of the subgraph mutation operator. The selected connected sub-tree is highlighted by thick edges and gray shape respectively.

graph G (lines 7 and 8). Finally, the resulting edge set E_s^* replaces the edges of E_T connecting the selected nodes from V_s. Note that the tree property is maintained by this operation, since a spanning sub-tree is replaced by another spanning sub-tree.

B. Example

Figure 5 illustrates the application of our mutation operator in an example. The figure shows the relevant part of a given candidate solution. We assume that $e_r = (c, d)$ was selected in line 1 as the starting edge (see Figure 5a). Hence, $V_s = \{c, d\}$. We assume $\sigma = 4$ (line 3). Since $|V_s| = 2 < 4$ the body of the loop (lines 4-5) is entered and all nodes incident to the nodes in V_s (which are not yet included) are added to V_s: $V_s = V_s \cup \{b, e\}$ (see Figure 5b). The loop terminates after this step since the termination criterion is met ($|V_s| > \sigma$). Next, a single objective is sampled (line 6) and sub-graph G' with node set V_s (line 7; Figure 5c) is passed to Prim's algorithm. This yields the MST on G' regarding the sampled objective. Finally, the sub-tree composed of edges E_s is replaced with edges E_s^* computed before (refer to Figure 5d).

C. Properties

Given a candidate solution T and a solution T', which is obtained by application of the sub-tree mutation to T we find that T' either dominates T or T' and T are incomparable. This is evident under the following considerations: let $o \in \{1, \ldots, m\}$ be the sampled objective (line 6 in algorithm 1). A sub-tree of T is replaced with the MST regarding objective o. Hence, $c_o(T') \leq c_o(T)$. If by chance $c_p(T') \leq c_p(T)$ for all other objectives $p \neq o$, then $T' \preceq T$. Otherwise the replaced sub-tree is incomparable und thus T and T' are incomparable, too. Consequently, T' cannot be dominated by T[6]. In objective space, this implies either a step towards the Pareto-front or a "side-step" supporting diversity (see Figure 6, left hand side,

[6]Note that T' is incomparable to T, but might be dominated by another solution from the population.

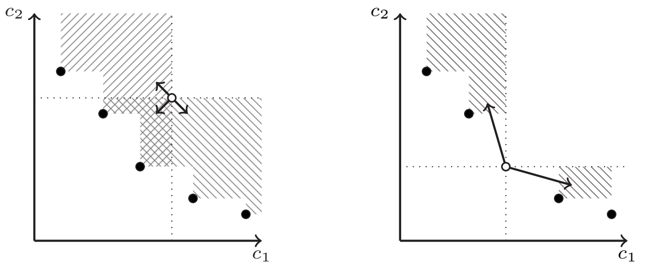


Fig. 6. Illustration of the objective space regions reachable by application of the sub-tree mutation operator on a non Pareto-optimal spanning tree (left) and a Pareto-optimal spanning tree (right).

for an illustration). If the mutation is applied to a Pareto-efficient solution (see Figure 6, right hand side), diversity is enforced by a side-step-behavior only.

As the operator drives convergence and diversity at the same time without allowing deterioration, we term it a *Pareto-beneficial* mutation. However, note that—dependent on the topology of the considered spanning tree—this mutation, in general, is not able to reach all solutions in decision space. This violates the important reachability design rule for general mutation operators [18]. Thus, in practice, the sub-tree mutation should be combined with a simple mutation or support an alternative mode, which ensures reachability. We will also consider this aspect during the following experiments.

V. Experiments

A. Problem instances

Consistent with [6] and [5], we consider bi-criteria mcMST instances. These instances are created using the generation process described in section III-A. We create complete graphs of size 10, 20, 30, 40, 50, and 100 nodes. Ten different instances with uniformly randomly distributed edge weights at all problem sizes are generated.

B. Method and Parameters

For evaluation we consider three different mutation operators: Zhou and Gen's uniform mutation on a Prüfer code representation, the edge exchange mutation on a direct encoding, and the proposed sub-tree mutation on direct encoding. Additionally, we also evaluate the combined (mixed) application of sub-tree and edge-exchange mutation. As the sub-tree mutation is principally biased to generate Pareto-beneficial solutions, it is unable to reach any solution in decision space. To compensate for this, we uniformly randomly decide for either applying sub-tree or edge-exchange mutation in a mixed scenario.

To allow unbiased and fair comparison, we apply two modern standard evolutionary multi-criteria optimization algorithms, namely NSGA-II [19] and SMS-EMOA [20] as encapsulating meta-heuristics, in which each mutation is tested. While NSGA-II relies on non-dominated sorting for convergence and on crowding distance as secondary selection criterion for diversity preservation, SMS-EMOA implements a steady-state approach, which replaces a single solution in the population based on hypervolume [21] contribution.

TABLE I
Experimental settings for all configurations of the applied algorithms NSGA-II and SMS-EMOA.

	Setting
Parameter	NSGA-II / SMS-EMOA
population size μ	100
# of offspring λ	100
# of evaluations	$1000 \cdot \|V\|$
# independent runs	10

To only observe mutation effects, we explicitly deactivate crossover operators for all algorithm configurations[7]. Evaluations are repeated 10 times for each instance. The quality of the Pareto-front is measured as the hypervolume enclosed by the non-dominated solutions and a reference point. The reference point is determined considering all the results of all algorithm configurations for a single instance.

The parameters used for algorithm configuration are given in Table I. The source code for instance generation as well as our implementations of the algorithms are bundled in the R-package **`mcMST`**[8] [17]. The generated instances are available at our project website[9].

C. Results

Figure 7 shows the hypervolume distribution gained by using the different operators split up by instance size. Obviously, the uniform mutation operator on the Prüfer encoding as used by Zhou and Gen performs worst in all cases. For small instances ($\leq$ 30 nodes), the proposed sub-tree mutation is outperformed by the edge-exchange operator. For larger instances, however, the sub-tree mutation significantly outperforms the edge-exchange mutation. The mixed scenario yields for all cases best hypervolume results. These observations confirm the advantageous properties of our proposed mutation operator for relevant problem sizes. The observation, that edge-exchange mutation alone is significantly outperformed regarding hypervolume by the sub-tree mutation—except for small instances—suggests that dominating and more diverse solutions are produced by the new operator.

The expected better convergence behavior as well as more diverse solution fronts can be observed in Figure 8 for a 100 node instance. We restrict the plot to NSGA-II results only. For SMS-EMOA, results are comparable.

While the uniform operator shows a less converged and little diverse approximation of the Pareto-front (yellow front), the edge-exchange operator leads to a better converged approximation (purple front). In contrast, the Pareto-front approximated under the influence of the proposed sub-tree mutation exposes

[7]We stress that our foremost goal is to evaluate mutation and to develop a beneficial mutation operator. Applying crossover in all algorithm configurations may be advantageous in practice. For evaluating mutation properties, it is certainly obstructive.

[8]https://github.com/jakobbossek/mcMST

[9]https://jakobbossek.github.io/mcMSTproject/

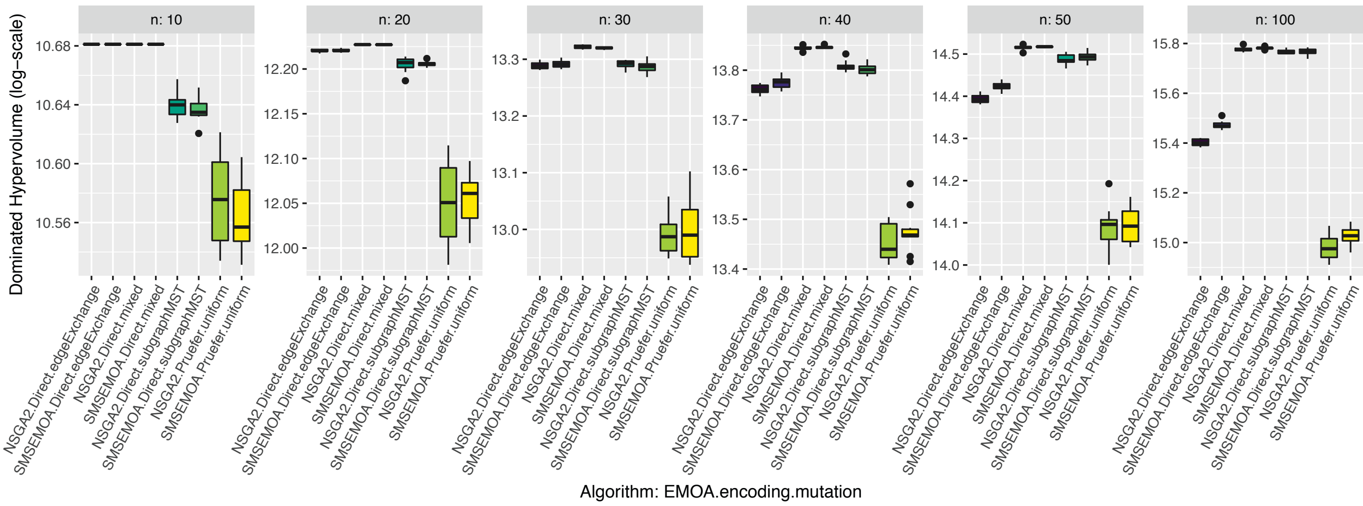


Fig. 7. Distribution of the dominated hypervolume (log-scale) for examplary instances of $|V| = 10, 20, 30, 40, 50, 100$ nodes respectively.

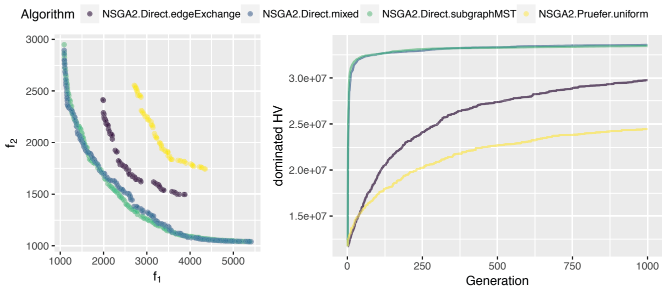


Fig. 8. Left: examplary approximations of the Pareto-front for all considered combinations of tree encodings and mutation operators. Right: the optimization trace of the dominated hypervolume resulting in the corresponding approximations in the left plot. Clearly, the sub-tree mutation with/without intermingled edge exchange advances to the Pareto-front rapidly, while the other mutations exhibit inferior performance.

high diversity and even better convergence behavior (green front). The same holds for the mixed application of sub-tree and edge-exchange mutation. This confirms the before discussed Pareto-beneficial properties: the sub-tree mutation supports convergence and diversity.

The overall gain in convergence speed and diversity generation is exemplarily visualized in Figure 8 (right). The proposed sub-tree mutation leads to an enormous speed-up in convergence and reaches high quality results in a small fraction of time compared to the remaining mutation variants. The same holds—due to the influence of the new mutation operator—also for the mixed scenario.

VI. Discussion and Conclusion

First and foremost, the results of our experiments prove the suggested sub-tree mutation operator being a valuable contribution to solve the mcMST problem with evolutionary algorithms. We introduced a Pareto-beneficial mutation operator which guarantees generating only dominating or incomparable solutions. This enables tendencies towards convergence and diversity together.

At the same time, this work can only be considered as a starting point for more and exciting investigations of the multi-criteria minimum spanning tree problem with special focus on variation operators. The currently proposed mutation is costly compared to the simple unbiased edge-exchange variant and is not able to reach all areas of decision space. Therefore, combining advanced but possibly costly operators, which speed up convergence and / or increase diversity, with computationally cheap, unbiased, and random operators that ensure reachability in decision space seems to be a reasonable approach. We showed that this can lead to overall beneficial solution quality. However, the exact proportion of using advanced and simple methods has to be investigated in depth. Moreover, investigations on the general applicability of the proposed mutation operator should be permormed on a more diverse set of problem instances with different Pareto-front shapes and variants of the mcMST problem itself.

Finally and equally important, our approach shows that properties of the mcMST problem (and also other problems, in general) should be considered in more detail. Instead of relying on unbiased and fully random operators like uniform position swaps or edge exchanges, operators should exploit known characteristics and even heuristic insights on problem classes. This may—like in this case—lead to a mutation operator that addresses both, convergence and diversity in an advantageous way and advance multi-criteria evolutionary optimization also on the level of variation operators.

References

[1] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957. [Online]. Available: http://dx.doi.org/10.1002/j.1538-7305.1957.tb01515.x

[2] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956.

[3] P. Camerini, G. Galbiati, and F. Maffioli, "On the complexity of finding multi-constrained spanning trees," *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 39 – 50, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0166218X83900148

[4] S. Ruzika and H. W. Hamacher, "Algorithmics of large and complex networks," J. Lerner, D. Wagner, and K. A. Zweig, Eds. Berlin, Heidelberg: Springer-Verlag, 2009, ch. A Survey on Multiple Objective Minimum Spanning Tree Problems, pp. 104–116. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02094-0_6

[5] J. D. Knowles and D. W. Corne, "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, 2001, pp. 544–551 vol. 1.

[6] G. Zhou and M. Gen, "Genetic algorithm approach on multi-criteria minimum spanning tree problem," *European Journal of Operational Research*, vol. 114, no. 1, pp. 141 – 152, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221798000162

[7] H. Prüfer, "Neuer Beweis eines Satzes über Permutationen." *Arch. Math. Phys.*, vol. 27, pp. 742–744, 1918.

[8] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994. [Online]. Available: http://dx.doi.org/10.1162/evco.1994.2.3.221

[9] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, Jun. 2000. [Online]. Available: http://dx.doi.org/10.1162/106365600568167

[10] J. Gottlieb, B. A. Julstrom, G. R. Raidl, and F. Rothlauf, "Prüfer numbers: A poor representation of spanning trees for evolutionary search," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO'01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 343–350. [Online]. Available: http://dl.acm.org/citation.cfm?id=2955239.2955297

[11] L. Han and Y. Wang, "A novel genetic algorithm for multi-criteria minimum spanning tree problem," in *Proceedings of the 2005 International Conference on Computational Intelligence and Security - Volume Part I*, ser. CIS'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 297–302. [Online]. Available: http://dx.doi.org/10.1007/11596448_43

[12] G. Chen, S. Chen, W. Guo, and H. Chen, "The multi-criteria minimum spanning tree problem based genetic algorithm," *Information Sciences*, vol. 177, no. 22, pp. 5050 – 5063, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025507002915

[13] P. Cardoso, M. Jesus, and Á. Márquez, "Multiple criteria minimum spanning trees," University of Algarve, Computing and Simulation Center CsC, Tech. Rep. 2006-01, 2006.

[14] D. Liu, K. Tan, S. Huang, C. Goh, and W. Ho, "On solving multiobjective bin packing problems using evolutionary particle swarm optimization," *European Journal of Operational Research*, vol. 190, no. 2, pp. 357 – 382, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037722170700567X

[15] C. Grimme, J. Lepping, and U. Schwiegelshohn, "Multi-criteria scheduling: an agent-based approach for expert knowledge integration," *Journal of Scheduling*, vol. 16, no. 4, pp. 369–383, 2013.

[16] A. Cayley, "A theorem on trees," *Quart. J. Math*, vol. 23, p. 376378, 1889.

[17] J. Bossek, "mcMST: A toolbox for the multi-criteria minimum spanning tree problem," *The Journal of Open Source Software*, vol. 2, no. 17, sep 2017. [Online]. Available: https://doi.org/10.21105/joss.00374

[18] H.-G. Beyer, *The Theory of Evolution Strategies*. Berlin: Springer, 2001.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[20] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, Eds., vol. 3410. Berlin Heidelberg: Springer, March 2005, pp. 62–76.

[21] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.