Solving Scalarized Subproblems within Evolutionary Algorithms for Multi-criteria Shortest Path Problems

Jakob Bossek(✉) and Christian Grimme

Information Systems and Statistics, University of Münster, Münster, Germany
{bossek,christian.grimme}@uni-muenster.de

Abstract. The $\mathcal{NP}$-hard multi-criteria shortest path problem (mcSPP) is of utmost practical relevance, e. g., in navigation system design and logistics. We address the problem of approximating the Pareto-front of the mcSPP with sum objectives. We do so by proposing a new mutation operator for multi-objective evolutionary algorithms that solves single-objective versions of the shortest path problem on subgraphs. A rigorous empirical benchmark on a diverse set of problem instances shows the effectiveness of the approach in comparison to a well-known mutation operator in terms of convergence speed and approximation quality. In addition, we glance at the neighbourhood structure and similarity of obtained Pareto-optimal solutions and derive promising directions for future work.

1 Introduction

Solving the shortest path problem plays a major role in tackling many network (flow) problems: When optimizing logistic networks, the routing of vehicles should ideally follow the shortest path from a starting location to a goal location. In computer networks, a shortest path may be the route with minimum latency. Luckily, with approaches from Dijkstra [9] to Floyd [13] there exist efficient approaches for finding optimal solutions - shortest paths from a starting location to all other or even between all location.

Often, however, problems are not restricted to finding a shortest path regarding a single objective. In logistic applications, costs for shortest paths may be computed based on distances, delivery time or fuel consumption (e. g., when considering the topology of a landscape and not only distances). Clearly, these objectives viewed separately may lead to different and contradicting solutions. Multi-objective optimization allows to consider contradicting objectives together and strives for a set of optimal compromises or Pareto-optimal solutions. These solutions cannot improve regarding one objective without deteriorating for another objective.

R. Battiti et al. (Eds.): LION 12 2018, LNCS 11353, pp. 184–198, 2019.
https://doi.org/10.1007/978-3-030-05348-2_17

The multi-criteria shortest path problem (mcSPP) is generally defined on a graph structure $G = (V, E, c)$, where V is the set of vertices, E is the set of edges, and $c : E \rightarrow \mathbb{R}^m_{>0}$ is a mapping of edges to an m-dimensional cost vector. Objective values for a path p through graph G result in an m-dimensional cost vector $F(p) = (f_1(p), \ldots, f_i(p), \ldots, f_m(p))^T \in \mathbb{R}^m_{>0}$ with $f_i(p) = \sum_{e \in p} c_i(e)$, when so-called sum objectives are considered.

The mcSPP is $\mathcal{NP}$-hard, as it exposes exponentially many Pareto-optimal solutions and is thus intractable, in general [10]. Solution strategies for this problem class vary from enumerative methods [14,15,20] through FPTAS [21,22] up to interactive methods [7] and randomized approaches like evolutionary algorithms [5,17,19]. The very popular enumerative approaches determine possible solutions by labelling or ranking. For two objectives, labelling methods work similar to the approaches in the single-objective case: each node can hold mutually non-dominating labels. After the labelling approach terminated, the labels at the target node represent the efficient set of solutions. Contrary, for two-objective problems, ranking approaches solve the single-objective k-shortest path problem by starting at the shortest path for one objective and extending k until all efficient solutions are found. Although, from the practical point of view, many mcSPP instances do not have an exponential number of solutions [18], enumerative methods strongly depend on the amount of solutions on the Pareto-front. In case of exponentially many solutions, these methods are not applicable any more.

Over the last years several authors applied evolutionary algorithms as general heuristic approaches for solving mcSPP. These randomized approximation methods follow the principle of Darwinian evolution by implementing a population-based evolutionary optimization loop, which comprises variation as explorative and selection as goal-directed promoting operator. However, for mcSPP, the application of multi-objective evolutionary algorithms can currently be considered as case-studies in which existing, general algorithmic concepts have merely been applied. Existing knowledge on the problem class has not been considered. Additionally, the existing approaches were only evaluated on few or even single instances.

In this work, we approach the mcSPP with multi-objective evolutionary algorithms using a diverse set of test instances to get empirically broader insight into approximation results. Following the principles in [3,4], we systematically review and improve mutation operators as algorithmic components for including beneficial problem knowledge in order to solve the shortest path problem under multiple criteria.

From here, the paper continues with a description of notation, methods, existing operators. Thereafter, we detail the proposed operators. As preparation for evaluation and discussion of results, we detail the generation process for used test instances. Finally, and after a careful discussion of results we conclude the paper.

2 Problem Formulation

Let $G = (V, E, c)$ be an undirected simple graph with node set $V = \{1, \ldots, n\}$, edge set $E = \{\{v, w\} \,|\, v, w \in V\}$ and vector-valued cost function $c : E \to \mathbb{R}^m_{>0}$, which associates each edge $e \in E$ with $m \geq 2$ positive weights, e. g., distance and travel costs in case of G representing a street network. A sequence $p_{v_i,v_j} = (v_i, v_{i+1}, \ldots, v_{j-1}, v_j)$ is called a $v_i - v_j$-path in G if $\{v_k, v_{k+1}\} \in E$ for $k = i, \ldots, j-1$. We define the i-th cost of a path p as the sum of i-th components of all edges lying on the path, i. e., $f_i(p) := \sum_{e \in p} c_i(e), i = 1, \ldots, m$. Wrapping up the path costs for all weights results in an m-dimensional cost vector $F(p) = (f_1(p), \ldots, f_m(p))^T \in \mathbb{R}^m_{>0}$. If $\mathcal{P}$ is the set of all feasible $s - d$-paths between a start node s and a destination node d the formulation

$$\begin{aligned} &F(p) = (f_1(p), \ldots, f_m(p))^T \to \text{min!} \\ &\text{s. t. } p \in \mathcal{P} \end{aligned}$$

defines the *multi-criteria shortest path problem* (mcSPP) with sum objectives[1]. In contrast to the single-objective case, where always an optimal (maybe not unique) solution exists, usually no single best solution can be found in the multi-criteria case. Instead, there exists a set of so-called non-dominated solutions $S = \{p \in \mathcal{P} \,|\, !\exists\, p' \in \mathcal{P} \text{ with } F(p') \preceq F(p)\}$. Here, the binary relation $\preceq$ is termed the *dominance relation* and is defined as follows: $p \preceq p'$, i. e., path p *dominates* path p', if $f_i(p) \leq f_i(p'), i = 1, \ldots, m$ and $\exists j \in \{1, \ldots, m\}, f_i(p) < f_i(p')$. In words: path p dominates another path p', if the cost vector of p is strictly better in at least one objective and not worse in the remaining objectives than the cost vector of p'. S is frequently termed *Pareto-set* and its image $F(S) \subset \mathbb{R}^m_{>0}$ *Pareto-front*. The goal is to approximate the Pareto-set and -front respectively. There exist classes of problems where $|S|$ is exponential in the number $n = |V|$ of nodes.

3 Considered Mutation Operators

Mutation operators have not received much attention in evolutionary multi-objective optimization for the mcSPP. A frequently adopted operator – termed random walk operator (RW) in the following – works as follows: Given a feasible $s-d$-path $p_{s,d} = (s = v_0, v_1, \ldots, v_l = d)$, select a random position $i \in \{0, \ldots, l-1\}$ and replace the subpath $p_{v_i,d} = (v_i, \ldots, v_l = d)$ with a random $v_i - d$-path (see Fig. 1 for an example). Since this operator does not use any information about the edge weights usually the probability to come up with a dominated path is quite high.

[1] Other objective types are possible, e. g., bottleneck objectives, but not considered in this work.

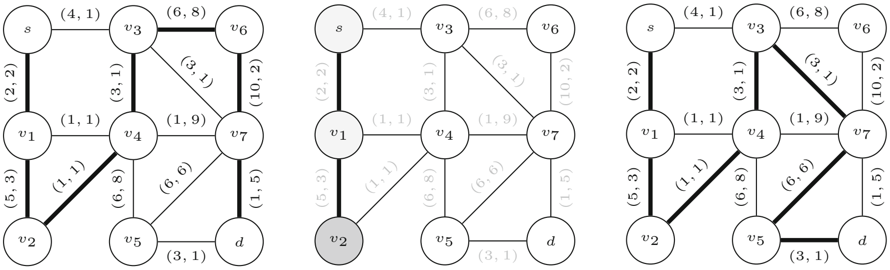

Fig. 1. Exemplary application of RW mutation. Left: initial path $p_{s,d}$ (bold edges). Center: select random position $v_i = v_2$, neglect all nodes located before v_2 in $p_{s,d}$ (gray nodes) and costs completely. Right: append random subpath $p_{v_2,d} = (v_2, v_4, v_3, v_7, v_5, d)$ to p_{s,v_2} (bold edges).

We propose the following alternatives which differ from RW by the method used to search for a new subpath. Here, we present the ideas for $m = 2$ objectives for sake of simplicity. However, adaptation to more than two objectives is straight-forward. Starting point is again a feasible path $p_{s,d} = (s = v_0, v_1, \ldots, v_l = d)$ and a randomly selected position $i \in \{0, \ldots, l-1\}$. Let p_{s,v_i} be the subpath from s to v_i. Instead of performing a random search, the first proposed *subgraph* operator (SG) selects a random cost component $o \in \{1, \ldots, m\}$ with equal probability, ignores the other cost components $\{1, \ldots, m\} \setminus \{o\}$ and searches for the shortest path from v_i to d with Dijkstra's algorithm. All nodes which are located prior to v_i on the input path $p_{s,d}$ are marked as visited to avoid loops. The resulting path $p_{v_i,d}$ is a minimal $v_i - d$-path regarding f_o. It is appended to p_{s,v_i} resulting in another feasible $s - d$-path $\tilde{p}_{s,d} = p_{s,v_i} \circ p_{v_i,d}$ where $\circ$ is the path concatenation. See Fig. 2 for an illustration of the working principle by example. Our second proposal is the *scalarized subgraph* operator (SGS) which is a generalization of SG. Instead of focusing on one of the objectives exclu-

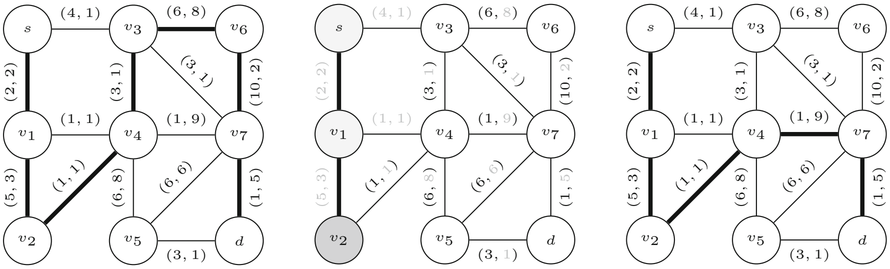

Fig. 2. Examplary application of subgraph mutation SG. Left: initial path $p_{s,d}$ (bold edges). Center: selected position is $v_i = v_2$. We sample the first objective, i. e., $o = 1$, ignore nodes on subpath p_{s,v_2} (gray nodes) and edge costs function $\neq o$. Right: find shortest $v_2 - d$-path $p_{v_2,d} = (v_2, v_4, v_7, d)$ with Dijkstra algorithm and append $p_{v_2,d}$ to p_{s,v_2} (bold edges).

sively, SGS samples a random weight $\lambda \in [0, 1]$, constructs the single-objective weighted sum problem $f_\lambda = \lambda f_1 + (1 - \lambda) f_2$ and solves this scalarized surrogate with a single-objective shortest path algorithm. As before, nodes already visited in p_{s,v_i} are marked visited. Again we end up with a $v_i - d$-path $p_{v_i,d}$, which can be appended to p_{s,v_i} resulting in a mutated $s - d$-path. An example is depicted in Fig. 3. Clearly, SG is a special case of SGS where $\lambda = 0$ or $\lambda = 1$ with equal probability.

The SGS mutation operator applies a weighted sum approach to subgraphs of G. Note, that weighted sum applied to G is only capable of finding so-called supported efficient solutions (see, e. g., [11]). These solutions are located on the convex hull of the Pareto-front. Solutions in concave regions cannot be detected with this approach. However, we expect, that applying weighted sum scalarization on subgraphs (ignoring already visited nodes) is able to (1) push solutions towards the Pareto-front rapidly and (2) identify solutions which are unreachable to the classic weighted sum approach. The first point is supported by the fact, that given a path p the path p' resulting from application of SG or SGS either dominates p or boths paths are incomparable. This is evident, since the appended subpath is a supported efficient solution of the shortest path problem on the subgraph and thus is non-dominated by any other possible subpath. We term this desirable behaviour *Pareto-beneficial* (see also [3]).

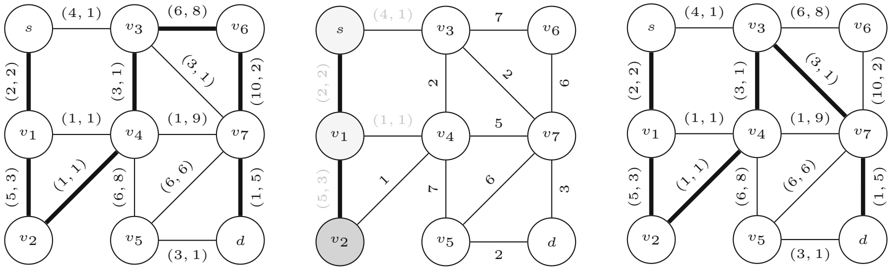

Fig. 3. Examplary application of SGS mutation. Left: Initial path $p_{s,d}$ (bold edges). Center: selected position is $v_i = v_2$. With the sampled weight (here $\lambda = 0.5$) we compute $f_\lambda(e) = \lambda f_1(e) + (1 - \lambda) f_2(e), \forall e \in E$ and ignore nodes located before v_i on the initial path (gray nodes). Right: find shortest $v_2 - d$-path regarding f_λ resulting in $p_{v_2,d} = (v_2, v_4, v_3, v_7, d)$ and append $p_{v_2,d}$ to p_{s,v_2} (bold edges).

4 Results

For an assessment of the proposed operators and their extensions, we conduct a series of experiments based on diverse graph topologies containing different amounts of nodes. Thus, we first detail the experimental setup and subsequently discuss the observations and results regarding this aspect. Afterwards, we briefly investigate solution properties based on the found solutions.

4.1 Graph Generation

In order to investigate the performance of the considered mutation operators we generated 150 random graphs in total: each 5 instances of 6 different topologies considering sizes $n \in \{50, 100, 250, 500, 1000\}$. The network topologies mimick real-world network structures and differ in network density, interconnection of nodes and (non)existence of clusters. The graph generation process is implemented in the R package `grapherator`[2] [2]. It follows a flexible three-step approach:

1. Place nodes in the Euclidean plane $[0, 100]^2$. Here we considered nodes placed uniformly at random within the bounding box or clustered nodes.
2. Establish links between nodes following different edge generation methods (complete graphs, edges based on Delauney triangulation or Waxmans probablistic model following [5]).
3. Associate each edge $e \in E$ with two additive uniform random weights $c_1(e) \in [20, 3000], c_2(e) \in [200, 5000]$. This last step is identical among all generated instances.

Figure 4 shows several generated networks by way of example.

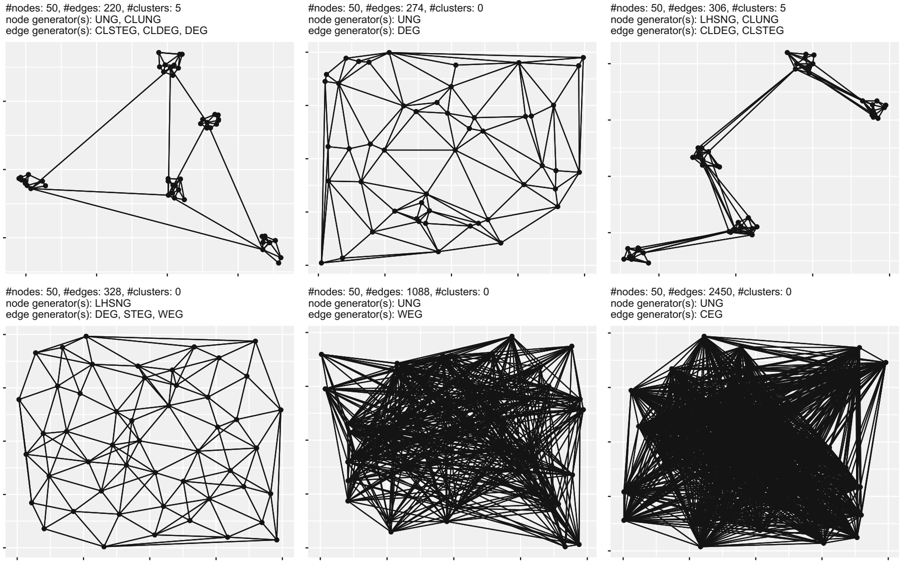
#nodes: 50, #edges: 220, #clusters: 5
node generator(s): UNG, CLUNG
edge generator(s): CLSTEG, CLDEG, DEG
#nodes: 50, #edges: 274, #clusters: 0
node generator(s): UNG
edge generator(s): DEG
#nodes: 50, #edges: 306, #clusters: 5
node generator(s): LHSNG, CLUNG
edge generator(s): CLDEG, CLSTEG
#nodes: 50, #edges: 328, #clusters: 0
node generator(s): LHSNG
edge generator(s): DEG, STEG, WEG
#nodes: 50, #edges: 1088, #clusters: 0
node generator(s): UNG
edge generator(s): WEG
#nodes: 50, #edges: 2450, #clusters: 0
node generator(s): UNG
edge generator(s): CEG

Fig. 4. Examplary graph topologies generated for our study.

[2] https://github.com/jakobbossek/grapherator.

4.2 Experimental Setup

We consider the three different mutation operators introduced in Sect. 2: random walk mutation (RW), subgraph mutation (SG) and scalarized subgraph mutation (SGS) as the generalization of SG. Two state-of-the-art evolutionary multi-objective algorithms are adopted as encapsulating meta-heuristics for each mutation operator. NSGA-II (non-dominated sorting genetic algorithm) [8] is a $(\mu + \lambda)$-strategy. It basically relies on non-dominated sorting as primary and crowding distance as secondary selection criterion. In contrast SMS-EMOA (S-metric selection) [12] – in its classical version – follows a $(\mu + 1)$ indicator based strategy. Here, the hypervolume [24] contribution of each individual is used directly for selection. Since our aim is the empirical investigation of mutation operators, recombination/crossover is not applied. However, we stress that the integration of recombination might be fruitful as well. All other parameters are wrapped up in Table 1. Each EMOA was executed 10 times for statistical soundness of subsequent performance assessment. We used the implementations from the R package `ecr` [1] as well as the packages' methods for performance assessment.

Table 1. Parameter settings for all configurations of the applied meta-heuristics NSGA-II and SMS-EMOA.

	Setting
Parameter	NSGA-II/SMS-EMOA
Population size μ	100
# of offspring λ	100/1
# of evaluations	$200 \cdot \lvert V \rvert$
# independent runs	10
Mutation prob. p_{mut}	1

In addition we used weighted sum scalarization (WSUM) [16] with 1000 equidistant weights $\lambda_k = \frac{k}{999}, k = 0, \ldots, 999$ and Dijkstra's algorithm for the single-objective shortest path problem to compute the supported efficient solutions as a baseline. Note that the drawback of this appealing approach is that it is not capable of finding non-supported Pareto-optima, i. e., solutions which are not located on the convex hull of the Pareto-front.

4.3 Performance of Mutation Operators

We start our observation of experimental data with considering different aggregated approximations of the Pareto-fronts in Fig. 5 for six graph instances with different topology, and sizes $n \in \{500, 1000\}$, respectively. The non-dominated solutions of all runs are aggregated for the application of the random walk (RW), subgraph (SG), and scalarized subgraph (SGS) mutation operators. In Fig. 5, we already qualitatively find that the SGS operator performs seemingly best in almost all cases. In this plot we also find an implicit further ranking of the other indicators: in many cases the SG operator is slightly superior to the random walk operator. Still, sometimes random walk mutation also produces good solution candidates in the final solution set.

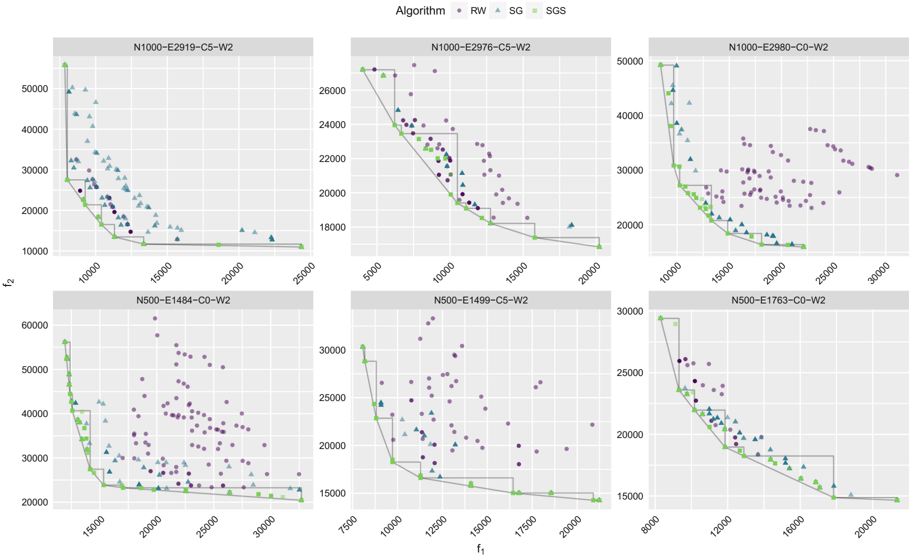

Fig. 5. Union of Pareto-front approximations of each 10 runs of all algorithms on some instances with $n = 500$ and $n = 1000$ nodes respectively. The instance names are encoded as follows: N<#nodes>-E<#edges>-C<#clusters>-W<#weights> where 0 clusters indicates unclustered, i. e., random, instances.

In Fig. 6 and Table 3, we analyze the observations systematically. Here, we restrict the analysis to NSGA-II results and the overall set of instances[3]. Note, that results for SMS-EMOA are congruent for the following observations. For two instances of each investigated number of vertices $n \in \{50, 100, 250, 500, 1000\}$ the distribution of hypervolume (HV) [24] values is given (top row of Fig. 6). This indicator measures convergence and diversity of solution sets by determining the enclosed volume of solutions and a reference point in objective space. Additionally, and considering the same instances, statistics for the cardinality of the final solution set (ONVG) [6] are given (bottom row of Fig. 6). For both indicators, the dashed line denotes the indicator values found by systematically iterated single-objective search using weighting of objectives (WSUM). This baseline is always

[3] Note, that we do not consider the topologies of generated instances due to space limitations. A detailed analysis of the topology's influence cannot be thought without considering the distribution of weights and the locations of start and destination notes. Therefore, this aspect is left for rigorous analysis in future work.

outperformed by the SGS operator, while the remaining operators sometimes stay below this baseline. Note, that the ONVG indicator is only meaningful, if it is considered secondary to the HV indicator. As a strongly dominated solution set can contain many (poor) solutions – often more, than a very good solution set – it is not sufficient as primary quality indicator. However, under the pre-condition of HV comparison, it enables us to discriminate solution quality in more detail. In order to get statistically sound quantitative results, we performed pairwise comparisons between the algorithms with the nonparametric Wilcoxon rank sum test on each test instance. We tested the hypothesis pair

$$H_0 : \text{med}(\text{HV}_A) \leq \text{med}(\text{HV}_B) \text{ vs. } H_1 : \text{med}(\text{HV}_A) > \text{med}(\text{HV}_B)$$

to check if the location shift between the hypervolume distributions HV_A and HV_B of algorithms A and B is significant at significance level $\alpha = 0.05$ adjusting the p-value with Bonferroni correction to avoid multiple-testing issues. Table 2 shows the aggregated test results. It turns out, that in the most interesting case (SGS vs. SG), the zero hypothesis is rejected in 109 of 150 cases ($\approx 73\%$ of the cases). In particular with growing instance size the number of rejected tests increases: H_0 is not rejected for 34 of 90 instances of size $n \in \{50, 100, 250\}$, but only for 7 out of 60 instances of size $n \in \{500, 1000\}$.

Thus, for the presented result, we can conclude that SGS does not only dominate with respect to hypervolume. It is also able to find the largest number of optimal solutions.

To complement the discussion of indicator values for the three different operators, we provide exemplary trajectories for two 500-nodes-instances on the HV development aggregated over all runs in Fig. 7. We find that both new operators SG and SGS converge very fast. After only few generations, the operators almost

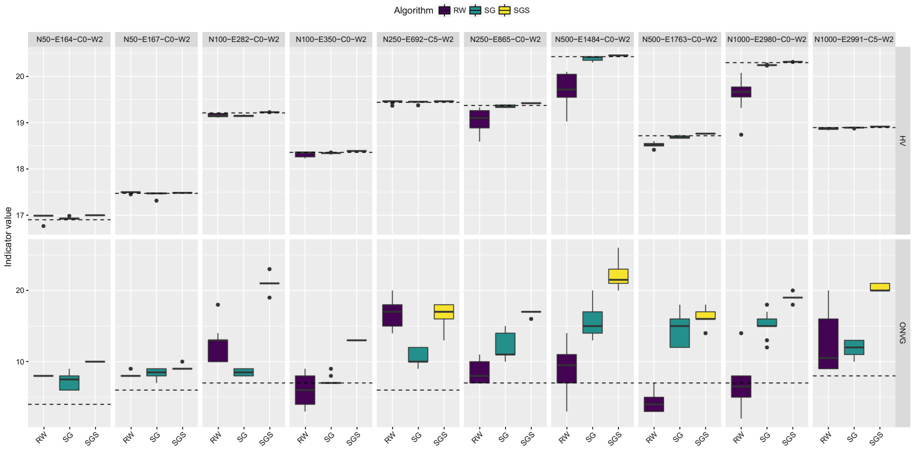

Fig. 6. Distributions of performance indicators HV (hypervolume on log-scale) and ONVG for each 2 exemplary instances of instance size $n \in \{50, 100, 250, 500, 1000\}$ respectively.

Table 2. Results of pairwise one-sided Wilcoxon rank sum test for location shift.

	H_0 not declined	H_0 declined
SG versus RW	96	54
SGS versus RW	47	103
SGS versus SG	41	109

reach their final HV value and thus the best approximation set. This speed-up is certainly related to the integration of local optimization on partial path. However, this speedup is also important in practise as the mutation operators SG and SGS are computationally more complex than the RW operator.

A final comparison of algorithm performance under different mutation operators is presented in Fig. 8. The shown heat-maps summarize the results of pairwise ε-indicator [24] evaluation. This indicator measures the degree of dominance between results of two algorithms A and B. It essentially states how far the result set of B has to be shifted towards the utopian point such that it is not dominated by A any more. Note, that this indicator is not symmetric. In other word, the larger the indicator for algorithm B compared to A and the smaller the indicator is for A compared to B, the better A performs compared to B. The results shown in Fig. 8 confirm the superiority of SGS over all other operators, the medium performance of SG, and the general low performance of the RW operator. For comparison reasons the WSUM approach for determining reference solutions is also included.

4.4 Properties of Solutions

To analyze the characteristics of solutions, we focus here on an exemplary instance comprising 500 nodes. We first create a scatter plot of the edge weights for this instance. Then we insert a frequency-based coloring of edges contained in solutions. As solutions we use the non-dominated set of solutions generated over all experimental runs for a given instance. Figure 9 shows the occurrence of edges in the solution set.

We can observe, that all (possibly) optimal solutions can be constructed from a subset of edges. This also means, that many edges are never part of (possibly) optimal solutions. At the same time, edges used in optimal solutions tend to gather in the lower left corner of the figure. Thus, edges with small costs in at least one component of the cost vector are more frequently used in (possibly) optimal solutions. The here exemplary presented observation is valid for all instances.

Table 3. Statistics of performance indicators for some exemplary instances.

Problem	Algorithm	avg.HV	sd.HV	avg.ONVG	sd.ONVG
N50-E164-C0-W21	RW	16.967	0.071	8.000	0.000
	SG	16.935	0.021	7.300	1.059
	SGS	**17.001**	0.000	**10.000**	0.000
N50-E167-C0-W24	RW	**17.493**	0.014	8.200	0.422
	SG	17.462	0.052	8.400	0.699
	SGS	17.486	0.004	**9.100**	0.316
N100-E282-C0-W24	RW	19.162	0.035	12.500	2.461
	SG	19.148	0.016	8.500	0.527
	SGS	**19.226**	0.000	**20.800**	1.135
N100-E350-C0-W25	RW	18.319	0.056	6.200	2.201
	SG	18.345	0.005	7.300	0.675
	SGS	**18.387**	0.000	**13.000**	0.000
N250-E692-C5-W21	RW	19.448	0.032	**16.800**	1.932
	SG	19.437	0.033	10.600	1.075
	SGS	**19.461**	0.003	16.600	1.713
N250-E865-C0-W25	RW	19.067	0.233	8.500	1.434
	SG	19.354	0.031	12.200	2.044
	SGS	**19.422**	0.002	**16.900**	0.316
N500-E1484-C0-W22	RW	19.724	0.361	9.000	3.399
	SG	20.391	0.045	15.600	2.366
	SGS	**20.453**	0.002	**22.200**	1.814
N500-E1763-C0-W25	RW	18.509	0.059	4.400	1.578
	SG	18.690	0.030	14.300	2.163
	SGS	**18.763**	0.001	**16.300**	1.059
N1000-E2980-C0-W23	RW	19.595	0.355	6.700	3.302
	SG	20.244	0.006	15.100	1.729
	SGS	**20.313**	0.001	**18.900**	0.568
N1000-E2991-C5-W23	RW	18.873	0.023	12.500	4.249
	SG	18.889	0.007	11.900	1.101
	SGS	**18.914**	0.001	**20.300**	0.483

Additionally, we use two visualizations based on the number of common edges (NCE) and the largest common sub-path (LCS) between two solutions. Per row/column, we visualize the comparison of a non-dominated solution with any other non-dominated solution of the solution set. The solutions are ordered according to objective value f_1. Thus, distant solutions in a row (column) are also distant in the non-dominated front; neighbouring solutions in a row (column) are also neighbours in the non-dominated front. The left-hand plot in Fig. 10

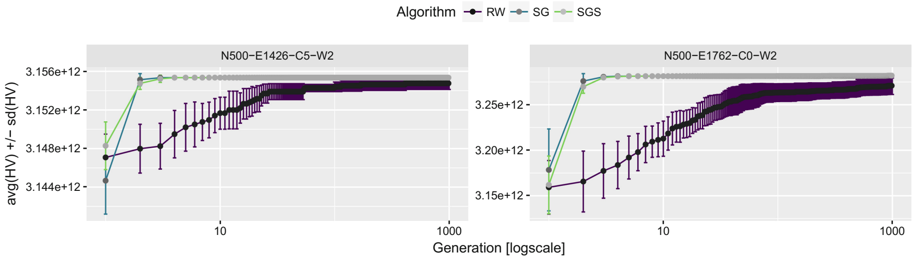

Fig. 7. Hypervolume trajectories (average of all runs ± standard devition for each EMOA for each generation) on two examplary graphs with 500 nodes.

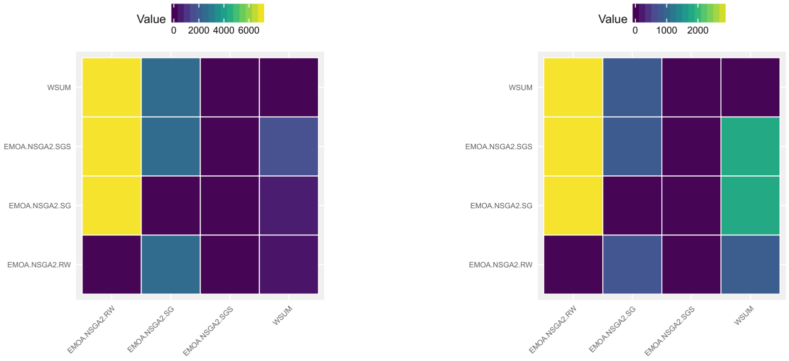

Fig. 8. Representative heat-maps of pairwise ε-indicator values for a problem with 500 nodes (left) and 1000 nodes (right).

shows the similarity of solutions regarding the NCE, while the right-hand plot shows the length of the largest common sub-path of two solutions, respectively.

In both visualizations we find, that there are only local neighbourhood relations between neighbouring solutions. Combined with the largest common sub-path behaviour (where the same effect can be observed) this means, that solutions can locally be transformed to neighbouring solutions by changing one or few edges. Interestingly, a little bit more distant solutions in objective space have almost nothing in common. As such, paths are completely disjunct for these solutions. That in turn suggests, that solutions cannot be constructed from each other by simply combining "optimal" building blocks. In fact, constructing new path (as it is done by the SGS operator, guided by a combined objective) can contribute to finding alternative non-dominated routes.

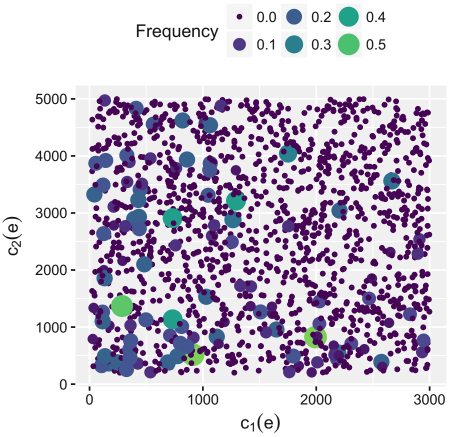

Fig. 9. Analysis of edge occurrence frequency for overall non-dominated $s-t$-paths in the solution sets of an examplary instance with $n = 500$ nodes. Scatter plot of edge weights. Each edge is sized/coloured by the fraction of non-dominated paths it is part of.

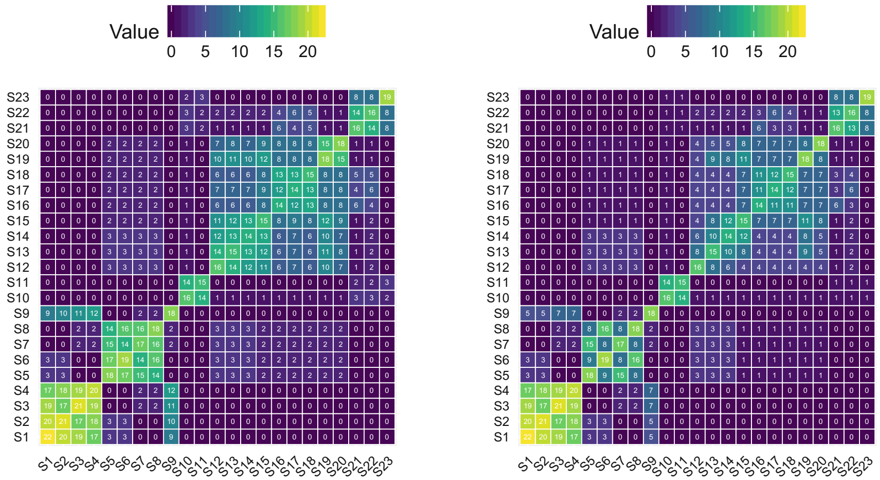

Fig. 10. Analysis of neighbourhood structure for overall non-dominated $s-t$-paths in the solution sets of an examplary instance with $n = 500$ nodes. Left: heat-map of pairwise NCE (number of common edges). Right: heat-map of pairwise size of largest common sub-path values.

5 Conclusions

This work deals with mutation in evolutionary algorithms for the multi-criteria shortest path problem. We present a mutation approach which follows the idea of replacing a subpath of an existing solution with another locally efficient subpath. Hence, the operator mimicks the well-known weighted sum scalarization approach in a smaller world and furthermore generates an offspring individual which is not dominated by its parent. A systematic comparison with simple random-path substitution and weighted sum scalarization reveals promising results. The new mutation operator does not only produce good approximations regarding dominated Hypervolume (statistically significant in most cases). It furthermore finds solutions in concave regions of the Pareto-front (which the global weighted sum approach is not capable of) and shows a rapid convergence behaviour.

Future work directions are manifold. We aim to test the proposed operator on specifically generated instances with a large number of non-dominated solutions. Furthermore, selection of the cutpoint in parent solutions is done uniformly at random at the moment. Here, we are confident, that smarter selection strategies may be beneficial in terms of even faster convergence. Additionally, multi-objective heuristics specifically designed for problem decomposition, such as MOEA/D [23], can be evaluated in conjunction with the new operators. Last but not least, theoretical analysis of the proposed mutation operator is desirable in order to understand its functionality in a more[4] rigorous way.

Acknowledgments. The authors acknowledge support from the *European Research Center for Information Systems (ERCIS).*

References

1. Bossek, J.: ecr 2.0: a modular framework for evolutionary computation in R. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2017, pp. 1187–1193 (2017). https://doi.org/10.1145/3067695.3082470
2. Bossek, J.: grapherator: a modular multi-step graph generator. J. Open Source Softw. **3**(22), 528 (2018). https://doi.org/10.21105/joss.00528
3. Bossek, J., Grimme, C.: A pareto-beneficial sub-tree mutation for the multi-criteria minimum spanning tree problem. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 3280–3287. IEEE, Honolulu (2017). https://doi.org/10.1109/SSCI.2017.8285183
4. Bossek, J., Grimme, C.: An extended mutation-based priority-rule integration concept for multi-objective machine scheduling. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 3288–3295. IEEE, Honolulu (2017). https://doi.org/10.1109/SSCI.2017.8285224
5. Chitra, C., Subbaraj, P.: Multiobjective optimization solution for shortest path routing problem. Int. Sch. Sci. Res. Innov. **4**(1) (2010)
6. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation, 2nd edn. Springer, New York (2007). https://doi.org/10.1007/978-0-387-36797-2
7. Coutinho-Rodrigues, J., Clmaco, J., Current, J.: An interactive bi-objective shortest path approach: searching for unsupported nondominated solutions. Comput. Oper. Res. **26**(8), 789–798 (1999). https://doi.org/10.1016/S0305-0548(98)00094-X

[4] https://www.ercis.org/.

8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolut. Comput. **6**(2), 182–197 (2002)
9. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**, 269–271 (1959)
10. Ehrgott, M., Gandibleux, X.: A survey and annotated bibliography of multiobjective combinatorial optimization. OR-Spektrum **22**(4), 425–460 (2000). https://doi.org/10.1007/s002910000046
11. Ehrgott, M., Gandibleux, X.: A survey and annotated bibliography of multiobjective combinatorial optimization. OR-Spektrum **22**(4), 425–460 (2000). https://doi.org/10.1007/s002910000046
12. Emmerich, Michael, Beume, Nicola, Naujoks, Boris: An EMO algorithm using the hypervolume measure as selection criterion. In: Coello Coello, Carlos A., Hernández Aguirre, Arturo, Zitzler, Eckart (eds.) EMO 2005. LNCS, vol. 3410, pp. 62–76. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31880-4_5
13. Floyd, R.W.: Algorithm 97: shortest path. Commun. ACM **5**(6), 345 (1962). https://doi.org/10.1145/367766.368168
14. Gandibleux, X., Beugnies, F., Randriamasy, S.: Martins' algorithm revisited for multi-objective shortest path problems with a maxmin cost function. 4OR **4**(1), 47–59 (2006). https://doi.org/10.1007/s10288-005-0074-x
15. Martins, E.Q.V.: On a multicriteria shortest path problem. Eur. J. Oper. Res. **16**(2), 236–245 (1984)
16. Miettinen, K.: Nonlinear Multiobjective Optimization. International Series in Operations Research & Management Science, vol. 12. Springer, New York (1998)
17. Mohamed, C., Bassem, J., Taicir, L.: A genetic algorithms to solve the bicriteria shortest path problem. Electron. Notes Discret. Math. **36**, 851–858 (2010). https://doi.org/10.1016/j.endm.2010.05.108. ISCO 2010 - International Symposium on Combinatorial Optimization
18. Müller-Hannemann, M., Weihe, K.: Pareto Shortest Paths is Often Feasible in Practice, pp. 185–197. Springer, Berlin (2001). https://doi.org/10.1007/3-540-44688-5_15
19. Pangilinan, J.M.A., Janssens, G.K.: Evolutionary algorithms for the multiobjective shortest path planning problem. In: International Journal of Computer and Information Science and Engineering, pp. 54–59 (2007)
20. Sanders, P., Mandow, L.: Parallel label-setting multi-objective shortest path search. In: 2013 IEEE 27th International Symposium on Parallel and Distributed Processing, pp. 215–224 (2013). https://doi.org/10.1109/IPDPS.2013.89
21. Tsaggouris, G., Zaroliagis, C.: Multiobjective Optimization: Improved FPTAS for Shortest Paths and Non-linear Objectives with Applications, pp. 389–398. Springer, Berlin (2006). https://doi.org/10.1007/11940128_40
22. Warburton, A.: Approximation of pareto optima in multiple-objective, shortest-path problems. Oper. Res. **35**(1), 70–79 (1987). https://doi.org/10.1287/opre.35.1.70
23. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evolu. Comput. **11**(6), 712–731 (2007)
24. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. Evolu. Comput. **7**(2), 117–132 (2003)