The Edge-Set Encoding Revisited: On the Bias of a Direct Representation for Trees

Carsten Tzoppe, Franz Rothlauf, and Hans-Josef Pesch

Working Paper 1/2004
January 2004

Working Papers in Information Systems

University of Mannheim
Department of Information Systems 1
D-68131 Mannheim/Germany
Phone +49 621 1811691, Fax +49 621 1811692
E-Mail: wifo1@uni-mannheim.de
Internet: `http://www.bwl.uni-mannheim.de/wifo1`

The Edge-Set Encoding Revisited: On the Bias of a Direct Representation for Trees

Carstten Tzoppe
Dept. of Information Systems 1
University of Mannheim
D-68131 Mannheim/Germany
carsten.tzschoppe@gmx.de

Franz Rothlauf
Dept. of Information Systems 1
University of Mannheim
D-68131 Mannheim/Germany
rothlauf@uni-mannheim.de

Hans-Josef Pesch
Department of Applied Mathematics
University of Bayreuth
95440 Bayreuth/Germany
hans-josef.pesch@uni-bayreuth.de

January 26, 2004

Abstract

The edge-set encoding is a direct tree representation which directly represents trees as sets of edges. There are two variants of the edge-set encoding: the edge-set encoding without heuristics, and the edge-set encoding with heuristics. An investigation into the bias of the edge-set encoding shows that the crossover operator of the edge-set encoding without heuristics is unbiased, that means it does not favor particular types of trees. In contrast, the crossover operator with heuristics is biased towards the simple minimum spanning tree (MST) and generates more likely trees that are MST-like. As a result, the performance of the edge-set encoding without heuristics does not depend on the structure of the optimal solution. Using the heuristic crossover operator results only in high GA performance if the optimal solution of the problem is slightly different from the simple MST. However, if the optimal solution is not very similar to the simple MST a GA using the heuristic crossover operator fails and is not able to find the optimal solution. Therefore, it is recommended that the edge-set encoding with heuristics should only be used if it is known a priori that the optimal solution is very similar to the simple MST. If this is not known a priori, other unbiased search operators and representations should be used.

1 Introduction

A spanning tree of an undirected graph G is a subgraph that connects all vertices of G and contains no cycles. Relevant constraint minimum spanning tree (MST) problems are, for example, the optimal communication spanning tree (OCST) problem, or the degree-constrained minimum spanning tree problem (Narula & Ho, 1980; Fekete, Khuller, Klemmstein, Raghavachari, & Young, 1997; Raidl &

Julstrom, 2003). The NP-hard OCST problem (Hu, 1974) seeks a spanning tree that connects all given nodes and satisfies their communication requirements for a minimum total cost. Genetic algorithms (GAs) have been applied with success to many constrained MST problems. As it is well known that the proper design of operators and representations is crucial for GA performance (Rothlauf, 2002), a large variety of tree representations like NetKeys (Rothlauf, Goldberg, & Heinzl, 2002), the link-and-node-biased encoding (Palmer & Kershenbaum, 1994), or Prüfer numbers (Prüfer, 1918; Gottlieb, Julstrom, Raidl, & Rothlauf, 2001) have been developed. Recently, Raidl and Julstrom (2003) proposed a new direct representation of trees, the edge-set encoding, which has successfully been used for the degree-constrained MST problem, and has outperformed other representations such as Prüfer numbers or NetKeys.

The purpose of this paper is to investigate the properties of the edge-set encoding. The paper focuses on the crossover operators, heuristic and non-heuristic KruskalRST*, used for the edge-set encoding and examines whether they are biased that means they overrepresent specific tree structures. Furthermore, the performance of the crossover operators is compared for OCST problems. The results show that the heuristic crossover operator is strongly biased towards the simple MST, whereas the non-heuristic crossover operator shows no bias. Consequently, GA performance increases when using the heuristic crossover operator for OCST problems where the optimal solution is only slightly different to the simple MST. In contrast, when applying the heuristic crossover operator to OCST problems where the optimal solution is not similar to the simple MST, GAs using the edge-set encoding fail.

The paper is structured as follows. The following section gives a short definition of the OCST problem and describes the functionality of the edge-set encoding with and without heuristics. Section 3 investigates the bias of the two variants of the crossover operator of the edge-set encoding. After investigating the bias, section 4 examines the influence of the crossover operator with and without heuristics on the performance of GAs when solving the OCST problem. The paper ends with concluding remarks.

2 Setting up the Stage: The OCST Problem and the Edge-Set Encoding

2.1 The OCST Problem

The optimal communication spanning tree problem (also known as minimum spanning tree problem or simple network design problem) was first introduced in Hu (1974):

Definition 1: Optimal Communication Spanning Tree Problem
Let $G = (V, E)$ be a complete undirected graph. $n = |V|$ denotes the number of nodes in the graph and $m = |E|$ denotes the number of edges in the graph. To every pair of nodes (i, j) a non-negative weight w_{ij} and a non-negative communication requirement r_{ij} is associated. The communication cost $c(T)$ of a

spanning tree T is defined as

$$c(T) = \sum_{i,j \in V,\, i<j} r_{ij} \cdot w(p^T_{i,j}),$$

where $w(p^T_{i,j})$ denotes the weight of the unique path from node i to node j in the spanning tree T. The OCST Problem seeks the spanning tree with minimal costs among all other spanning trees.

Definition 2: Minimum Spanning Tree Problem
The OCST problem becomes the minimum spanning tree (MST) problem if there are no communication requirements r_{ij} and $c(T)$ depends only on the weights w_{ij}. Then, T is the simple minimum spanning tree if $c(T) \leq c(T')$ for all other spanning trees T', where $c(T) = \sum_{(i,j)\in T} w_{ij}$.

The similarity between two spanning trees T_i and T_j can be measured using the distance $d_{ij} \in \{0, 1, \ldots, n-1\}$ as $d_{ij} = \frac{1}{2}\sum_{u,v\in V,\, u<v} |l^i_{uv} - l^j_{uv}|$, where l^i_{uv} is 1 if a link from u to v exists in T_i and 0 if it does not exist in T_i.

Like many other constrained spanning tree problems, the OCST problem is NP-hard (Garey & Johnson, 1979). Even more, it was later shown that the OCST problem is $\mathcal{MAX}\ \mathcal{SNP}$-hard (Papadimitriou & Yannakakis, 1991), that means it cannot be solved using a polynomial-time approximation-scheme, unless $\mathcal{P} = \mathcal{NP}$. Nevertheless, the OCST problem has been studied extensively in the literature and many researchers have tried to develop efficient approximation algorithms. The current best approximation algorithm for the general OCST problem approximates the optimal solution with $c(T) = O(\log n \log\log n) \cdot c(G)$ (Charikar, Chekuri, Goel, Guha, & Plotkin, 1998), where $c(G)$ is the cost of the network when using G. $c(G)$ is a lower bound for the cost of a spanning tree $c(T)$ as the weight of the unique path between node i and j in a spanning tree T is greater or equal in comparison to the weight of the path with minimal weight connecting the nodes i and j in G. As there are no efficient approximation algorithms, many researchers used GAs for solving the OCST problem (Palmer, 1994; Berry, Murtagh, & McMahon, 1995; Li & Bouchebaba, 1999; Kim & Gen, 1999; Rothlauf, 2002; Chou, Premkumar, & Chu, 2001).

It was shown in Rothlauf, Gerstacker, and Heinzl (2003) that the optimal solution of a OCST problem is similar to the simple MST, that means the average number of different edges between the OCST and the simple MST is significantly lower than the average number of different edges between a randomly generated tree and the simple MST. Therefore, as the optimal solution of an OCST problem is biased towards the simple MST, representations as well as operators that favor or overrepresent trees which are similar to the MST are expected to solve the OCST problem more efficiently.

2.2 The Edge-Set Encoding without Heuristics

The edge-set encoding (Raidl & Julstrom, 2003) is a direct tree representation which means it directly represents trees as sets of edges. In the following paragraphs we describe how initial populations are created and explain the functionality of the crossover and mutation operator of the edge-set encoding without heuristics.

Initialization: In order to create feasible solutions for the initial population, the edge-set encoding uses the Kruskal random spanning tree (RST) algorithm, a slightly modified version of the algorithm from Kruskal. In contrast to Kruskals' algorithm , KruskalRST chooses edges (i, j) not according to their weight w_{ij} but randomly. Raidl and Julstrom (2003) have shown that this algorithm, KruskalRST, for creating random spanning trees has a small bias towards star-like trees.

```
procedure KruskalRST($V, E$):
$T \leftarrow \emptyset,\ A \leftarrow E$;
while $|T| < |V| - 1$ do
  choose an edge $\{(u, v)\} \in A$ at random;
  $A \leftarrow A - \{(u, v)\}$;
  if $u$ and $v$ are not yet connected in $T$ then
    $T \leftarrow T \cup \{(u, v)\}$;
return $T$.
```

Raidl and Julstrom (2003) also presented two other RST algorithms (PrimRST, RandWalkRST) for generating the initial population, but RandWalkRST has an unlimited worst-case running time, and PrimRST has a stronger bias in comparison to KruskalRST.

Recombination: The functionality of the crossover operator is straightforward. To obtain an offspring T_{off} from two parental trees T_1 and T_2, KruskalRST is applied to the graph $G_{cr} = (V, T_1 \cup T_2)$. Therefore, the resulting crossover operator has high heritability as in the absence of constraints, only parental edges are used to create the offspring. Crossover becomes more complicated for constraint MST problems as it is possible that the RST algorithm can create no feasible tree from $G_{cr} = (V, T_1 \cup T_2)$. Then, additional edges have to be chosen randomly to complete an offspring. Based on KruskalRST, Raidl and Julstrom (2003) distinguished two different recombination operators: The variant previously described is denoted KruskalRST crossover. The second variant is denoted KruskalRST* crossover. When using this variant, edges that are common to both parents T_1 and T_2, are included in the offspring T_{off} before KruskalRST crossover is applied. Results from Raidl and Julstrom (2003) indicated a better performance of the KruskalRST* crossover for the degree-constraint MST problem.

Mutation: The mutation operator randomly replaces one edge in the spanning tree. This replacement can be implemented in two ways. The first variant of the mutation operator chooses randomly one edge from $E \setminus T$ and includes it in T. This creates a cycle. Then, the operator randomly chooses one edge of the cycle and removes it from T ("insertion before deletion"). The second variant first randomly deletes one edge from T and connects then the two disjunct connected components using a random edge from $E \setminus T$ ("deletion before insertion").

2.3 The Edge-Set Encoding with Heuristics

The following paragraphs describe how heuristics that rely on the weights w_{ij} can be included in the edge-set encoding.

Heuristic Initialization: To favor low-weighted edges when generating the initial population, the algorithm KruskalRST starts with sorting all edges in the underlying graph according to their weights w_{ij} in ascending order. The first spanning tree is created by choosing the first edges in the ordered list. As these are the edges with lowest weights, the first generated spanning tree is a simple MST. Then, the k edges with lowest weights are permuted randomly and more spanning trees are created again using the first edges in the list. Therefore, the heuristic initialization results in a strong bias of the initial population towards the simple MST. With increasing k the bias of the randomly created trees towards the simple MST is reduced. The number of edges, which are permuted, increases according to

$$k = \alpha(i-1)n/N,$$

where N denotes the population size, i is the number of the tree that is actually generated ($i = 1 \dots N$) and α is a parameter that controls the strength of the heuristic bias.

Heuristic Recombination: The heuristic recombination operator is a modified version of KruskalRST* crossover. Firstly, the operator transfers all edges $T_1 \cap T_2$ that exist in both parents to the offspring. Then, the remaining edges are chosen randomly from $E' = (T_1 \cup T_2) \setminus (T_1 \cap T_2)$ using a tournament with replacement of size two. If the underlying optimization problem is constrained, it is possible that the offspring has to be completed using edges not in E'.

Heuristic Mutation: The heuristic mutation operator is based on mutation by "insertion before deletion". In a pre-processing step all edges in the underlying graph are sorted according to their weights in ascending order. Doing this, a rank is assigned to every edge. To favor low-weighted edges, edges are not chosen randomly but according to their ranks

$$R = \lfloor |\mathcal{N}(0, \beta n)| \rfloor \bmod m + 1$$

where $\mathcal{N}(0, \beta n)$ is the normal distribution with mean 0 and standard deviation βn. β is a parameter that controls the bias towards low-weighted edges.

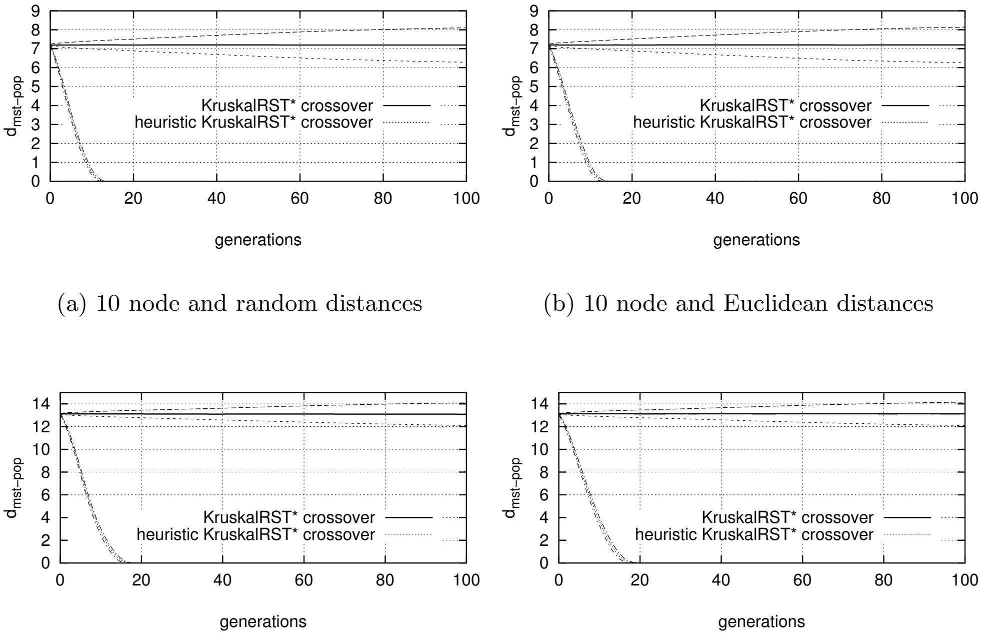
3 Investigating the Bias of the Edge-Set Encoding

A representation is unbiased if all possible solutions of the search space are represented uniformly (Rothlauf, 2002). Consequently, a search operator is unbiased if it does not overrepresent specific solutions, and the application of the search operator alone does not modify the statistical properties of a population. An unbiased search operator allows a uniform, non-directed search through the search space. A biased representation resp. operator should only be used if it is known a priori that the optimal solution of the underlying optimization problem is similar to the overrepresented solutions (Rothlauf & Goldberg, 2003). In contrast, unbiased representations resp. operators should be used if no a priori problem-specific knowledge is available. Then, the probability of finding the optimal solution is independent of the structure of the optimal solution.

To investigate if the crossover operator of the edge-set encoding with or without heuristics leads to an overrepresentation of MST-like individuals, we randomly generate an initial population with 500 individuals and apply the crossover operator iteratively. As no selection operator is used, no selection pressure exists to push the population to high-quality solutions. The crossover operator is unbiased if the statistical properties of the population do not change by applying crossover alone. In our experiments we measure in each generation the average distance $d_{mst-pop} = 1/N \sum_{i=1}^{n} d_{i,MST}$ of the individuals T_i in the population to the simple MST. If $d_{mst-pop}$ decreases, the crossover operator is biased towards the simple MST. In contrast, if $d_{mst-pop}$ remains constant, the crossover operator is unbiased and no solutions of the search space are overrepresented.

To obtain meaningful results, we performed this experiment on 50 randomly generated ten and 16 node problem instances with random, resp. Euclidean weights w_{ij}. For every problem instance we performed 50 runs with different, randomly chosen initial populations. In each run, the crossover operator is applied 100 times (generations). The communication requirements of the problem instances with random and Euclidean weights w_{ij} are equally distributed real values from $[0, 100]$. The random distance weights w_{ij} are real values and equally distributed from $[0, 100]$. When using Euclidean weights, all nodes are randomly placed on a 1000×1000 grid and the Euclidean distances between the nodes are taken as weights. As the weights w_{ij} are randomly created and $w_{ij} \neq w_{kl}$, $\forall i \neq l, j \neq l$, there is an unique optimum MST for every problem instance and distances to the simple MST are unique.

Figure 1 shows the mean and the standard deviation of the distance $d_{mst-pop}$ between the individuals in a population towards the simple MST over the number of generations for ten (top) and 16 (bottom) problem instances. The plots compare the non-heuristic KruskalRST* crossover with the heuristic KruskalRST* crossover operator (no selection is used). The results reveal that the crossover operator without heuristics is unbiased and does not modify the statistical properties of the population ($d_{mst-pop}$ remains constant over the number of generations). In contrast, the crossover operator with heuristics shows a strong bias towards the simple MST and the population converges quickly to the simple MST.



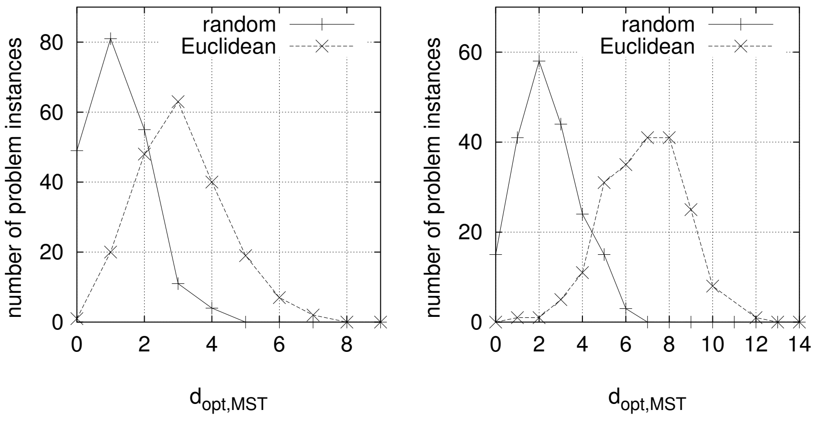
(a) 10 node and random distances

(b) 10 node and Euclidean distances

(c) 16 node and random distances

(d) 16 node and Euclidean distances

Figure 1: The plots show the mean and the standard deviation of the distance $d_{mst-pop}$ between a population of 500 randomly generated individuals towards the simple MST over the number of generations when using crossover only (no selection pressure). Results are presented for 10 (top) and 16 (bottom) problem instances using either random (left) or Euclidean (right) weights w_{ij}. The results show that the non-heuristic KruskalRST* crossover is unbiased that means the average distance between the population and the simple MST remains constant. The crossover operator with heuristics shows a strong bias towards the simple MST and a population converges to the simple MST in a few generations.



(a) 10 node

(b) 16 nodes

Figure 2: We randomly generated 200 OCST problems and show the distribution of the problem instances over the distance $d_{opt,MST}$ between the best found solution and the simple MST when using either random or Euclidean distance weights for ten (left) and 16 (right) node problems. The plots show that the optimal solutions for OCST problems using random distances weights are stronger biased in comparison to using Euclidean distances.

4 The Performance of the Edge-Set Encoding for the OCST problem

4.1 Finding Optimal Solutions for OCST Problems

To investigate how the performance of the edge-set encoding depends on the structure of the optimal solution, an optimal or near-optimal solution must be determined. The following experiments that should identify optimal or near-optimal solutions for OCST problems are similar to the ones described in Rothlauf, Gerstacker, and Heinzl (2003). They examined the OCST problem and showed that optimal solutions of OCST problems are biased towards the simple MST.

To determine the optimal (or near optimal) solution we apply a GA n_{iter} times to an OCST problem using a population size of N_0. T_0^{best} denotes the best solution of cost $c(T_0^{best})$ that is found during the n_{iter} runs. In a next round we double the population size and again apply a GA n_{iter} times with a population size of $N_1 = 2 * N_0$. T_1^{best} denotes the best solution with cost $c(T_1^{best})$ that can be found in the second round. We continue these iterations and double the population size $N_i = 2N_{i-1}$ until $c(T_i^{best}) = c(T_{i-1}^{best})$. This means we stop if the cost of the best solution T_i^{best} found in round i equals the cost of the best solution T_{i-1}^{best} found in round $i-1$. We assume that the solutions found using this approach are optimal or near-optimal.

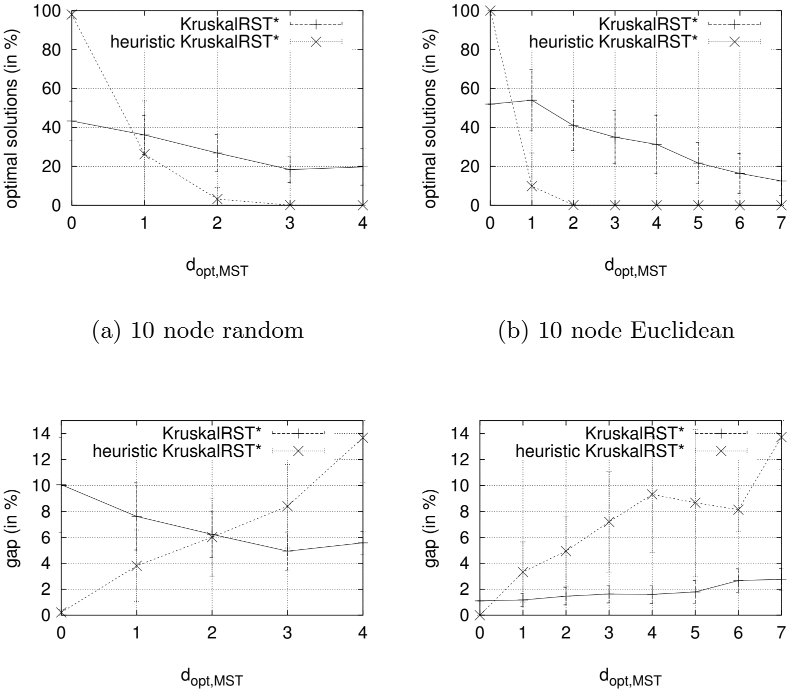
Figure 2 presents the results of our experiments. We show the number of problem instances over the distance $d_{opt,MST}$ between the best found solution and the simple MST for randomly created ten (Fig. 2(a)) and 16 (Fig. 2(b)) node OCST problem instances. We distinguish between random and Euclidean weights w_{ij}. For every problem instance we randomly generated 200 OCST problem instances. We used an initial population size $N_0 = 20$ for the ten node problem instances and $N_0 = 100$ for the 16 node problem instances, $n_{iter} = 20$, a standard GA with a NetKey representation (Rothlauf, Goldberg, & Heinzl, 2002), tournament selection without replacement of size two, uniform crossover and no mutation. The results are similar to the ones presented in Rothlauf, Gerstacker, and Heinzl (2003) and show that the best found solution is strongly biased towards the simple MST. Furthermore, OCST problems using random distance weights show a stronger bias in comparison to OCST problems using Euclidean distances.

4.2 Comparing Heuristic and Non-heuristic Crossover Operators

After determining optimal or near-optimal solutions as described in the previous paragraphs we examine the performance of the edge-set encoding on these 200 problem instances. We use the same randomly generated problem instances as in section 4.1 and investigate how the performance of the edge-set encoding using different types of crossover operators depends on the distance $d_{opt,MST}$ between the optimal (or near-optimal) solution and the simple MST. For comparing the performance of the two crossover operators, KruskalRST* and heuristic KruskalRST*, we use a standard GA with no mutation and tournament selection without replacement of size two. The initial population is generated using the non-heuristic KruskalRST (compare Sect. 2.2). Each run is stopped after the population is fully converged or the number of generations exceeds 200. We perform 100 runs for each of the 200 problem instances.

The population size N is chosen with respect to the performance of the crossover operator without heuristics (KruskalRST*). The aim is to find the optimal solution with a probability of about 50 %. Therefore, we choose for the ten node problems a population size of $N = 60$ (random weights), resp. $N = 100$ (Euclidean weights) and for the 16 node problems a population size of $N = 200$ (random weights) resp. $N = 450$ (Euclidean weights).

The results of our experiments are presented in Figure 3 (ten nodes) and Figure 4 (16 nodes). We show results for random (left) and Euclidean (right) weights. The top of Figure 3 and 4 shows the percentage of GA runs that correctly identify the optimal solutions at the end of a run over the distance $d_{opt,MST}$ between the optimal solution (compare section 4.1) and the simple MST. The bottom of the figures show the gap, $\frac{c(T_{found})-c(T_{opt})}{c(T_{opt})}$ (in percent), between the cost of the best found solution and the cost of the optimal solution that was identified in section 4.1 over $d_{opt,MST}$. The results reveal that the heuristic crossover operator, heuristic KruskalRST*, always finds the optimal solution if the optimal solution is the simple MST ($d_{opt,MST} = 0$). However, with increasing $d_{opt,MST}$ GA performance is significantly reduced and for $d_{opt,MST} \geq 3$ the optimal solution can not be found any more. In contrast, the



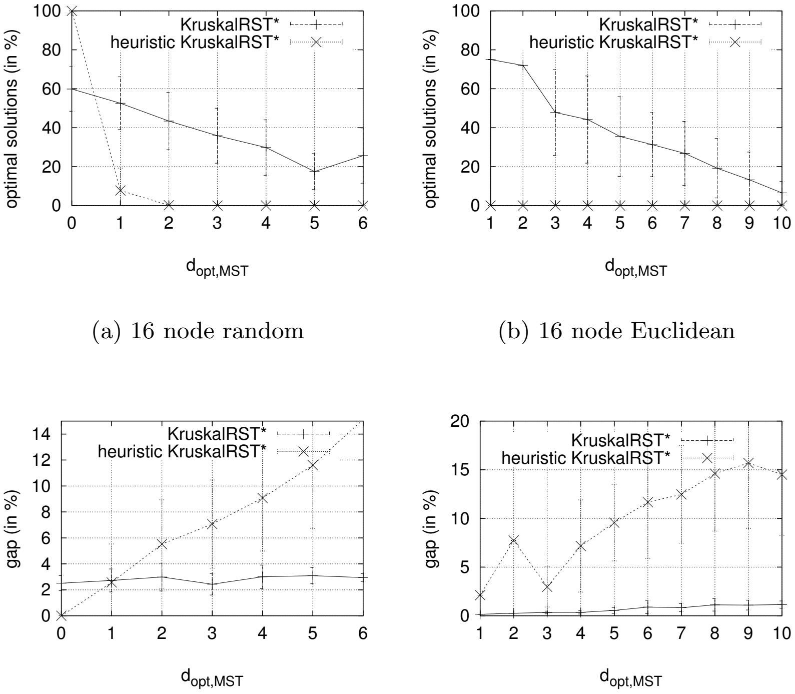
(a) 10 node random

(b) 10 node Euclidean

(c) 10 node random

(d) 10 node Euclidean

Figure 3: The figures compare the performance of different crossover operators for the edge-set encoding (KruskalRST* versus heuristic KruskalRST*) for randomly generated ten node OCST problem instances. The plots 3(a) and 3(b) show the mean and standard deviation of the percentage of optimal solutions that can be found over $d_{opt,MST}$. The plots 3(c) and 3(d) show the mean and standard deviation of the gap between the cost of the best found solution and the cost of the optimal solution determined in section 4.1. The results are averaged over 200 randomly created OCST problems using either random (left) or Euclidean (right) weights. The plots show that the heuristic KruskalRST* crossover outperforms the non-heuristic version if the optimal solution is very similar to the simple MST ($d_{opt,MST} \approx 0$). If the difference between the optimal solution and the simple MST becomes greater the heuristic KruskalRST* crossover results in low GA performance and is not able to find the optimal solution. In contrast, when using the non-heuristic KruskalRST* crossover, GA performance remains about constant with increasing $d_{opt,MST}$.



(a) 16 node random

(b) 16 node Euclidean

(c) 16 node random

(d) 16 node Euclidean

Figure 4: The figures compare the performance of different crossover operators for randomly generated 16 node OCST problem instances. The plots 4(a) and 4(b) show the mean and standard deviation of the percentage of optimal solutions that can be found over $d_{opt,MST}$. The plots 4(c) and 4(d) show the mean and standard deviation of the gap between the cost of the best found solution and the cost of the optimal solution determined in section 4.1. The results are averaged over 200 randomly created OCST problems using either random (left) or Euclidean (right) distance weights. The plots show that the heuristic KruskalRST* crossover outperforms the non-heuristic KruskalRST* crossover only if the optimal solution is very similar to the simple MST ($d_{opt,MST} \approx 0$).

performance of the crossover operator without heuristics decreases only slightly with larger $d_{opt,MST}$ and allows the GA to correctly identify the optimal solution even for larger $d_{opt,MST}$.

The direct comparison between the performance of the two crossover operators reveals that the heuristic crossover performs well only for problems where the optimal solution is slightly different from the simple MST. Otherwise, GAs using the edge-set encoding with heuristic crossover fail. These results are confirmed when examining the gap $\frac{c(T_{found})-c(T_{opt})}{c(T_{opt})}$ (bottom of Figure 3 and 4). Heuristic crossover shows perfect performance if the optimal solution is the simple MST. However, with increasing $d_{opt,MST}$ the quality of the solutions strongly decreases and the non-heuristic KruskalRST* outperforms the heuristic variant.

5 Summary and Conclusions

This work investigates different crossover variants, heuristic KruskalRST* crossover versus KruskalRST* crossover, of the edge-set encoding, which was proposed by Raidl and Julstrom (2003). Section 2 defines the optimal communication spanning tree (OCST) problem and describes the functionality of the edge-set encoding. In section 3 an investigation into the bias of the crossover operators is performed. Based on an analysis of optimal solutions for randomly generated instances of the OCST problem, section 4.2 investigates how the performance of the crossover operators of the edge-set encoding depends on the similarity between the optimal solution for an OCST problem and the simple minimal spanning tree (MST).

The investigation into the bias of the crossover operators for the edge-set encoding reveals that the heuristic KruskalRST* crossover is strongly biased towards the simple MST. In contrast to the unbiased, non-heuristic KruskalRST*, which results in a uniform search through the search space, the population converges quickly towards the simple MST if heuristic KruskalRST* crossover is used. Therefore, due to the strong bias towards the simple MST, GAs using the edge-set encoding with the heuristic KruskalRST* crossover can easily solve OCST problems if the optimal solution is the simple MST. However, with decreasing similarity between the optimal solution of an OCST problem and the simple MST, the edge-set encoding with heuristics fails as heuristic search gets stuck at the simple MST. In contrast, GAs using the edge-set encoding with the unbiased KruskalRST* crossover operator show good performance for all different OCST problems independently of the similarity between the optimal solution and the MST. The results suggest that the edge-set encoding with the heuristic KruskalRST* crossover operator is not appropriate for solving OCST problems. This search operator can only be used successfully if the optimal solutions are the simple MST, or slightly different variants of it.

The problems of the heuristic crossover operator of the edge-set encoding emphasizes the difficulty of a proper design of representations and operators. Especially the design of direct representations is difficult as in contrast to indirect representations, the behavior of new, problem-specific search operators is often unknown. The analysis of the edge-set encoding has shown that although optimal solutions for the OCST problems are biased towards the simple MST (Rothlauf, Gerstacker, & Heinzl, 2003), direct representations resp. operators like the heuristic KruskalRST* that use this problem-specific knowledge and are biased towards the simple MST, can fail in solving most of the randomly created OCST problem instances. Therefore, the authors recommend the use of unbiased representations if no problem-specific knowledge is known a priori. Proper representations for tree problems are for example non-heuristic versions of the edge-set encoding or NetKeys (Rothlauf, Goldberg, & Heinzl, 2002). In the case that biased representations resp. operators are used, it must be confirmed that the bias of the search fits to the properties of the optimal solutions. Otherwise failure is unavoidable.

References

Berry, L. T. M., Murtagh, B. A., & McMahon, G. (1995). Applications of a genetic-based algorithm for optimal design of tree-structured communication networks. In *Proceedings of the Regional Teletraffic Engineering Conference of the International Teletraffic Congress* (pp. 361–370). Pretoria, South Africa.

Charikar, M., Chekuri, C., Goel, A., Guha, S., & Plotkin, S. (1998, November). Approximating a finite metric by a small number of tree metrics.

Chou, H., Premkumar, G., & Chu, C.-H. (2001, June). Genetic algorithms for communications network design - an empirical study of the factors that influence performance. *IEEE Transactions on Evolutionary Computatios*, *5*(3), 236–249.

Fekete, S., Khuller, S., Klemmstein, M., Raghavachari, B., & Young, N. (1997). A network-flow technique for finding low-weight bounded-degree spanning trees. *Journal of Algorithms*, *24*, 310–324.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York: W. H. Freeman.

Gottlieb, J., Julstrom, B. A., Raidl, G. R., & Rothlauf, F. (2001). Prüfer numbers: A poor representation of spanning trees for evolutionary search. In Spector, L., Goodman, E., Wu, A., Langdon, W. B., Voigt, H.-M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M., & Burke, E. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference 2001* (pp. 343–350). San Francisco, CA: Morgan Kaufmann Publishers.

Hu, T. C. (1974, September). Optimum communication spanning trees. *SIAM Journal on Computing*, *3*(3), 188–195.

Kim, J. R., & Gen, M. (1999). Genetic algorithm for solving bicriteria network topology design problem. In Angeline, P. J., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A., & Porto, W. (Eds.), *Proceedings of the 1999 IEEE Congress on Evolutionary Computation* (pp. 2272–2279). IEEE Press.

Li, Y., & Bouchebaba, Y. (1999). A new genetic algorithm for the optimal communication spanning tree problem. In Fonlupt, C., Hao, J.-K., Lutton, E., Ronald, E., & Schoenauer, M. (Eds.), *Proceedings of Artificial Evolution: Fifth European Conference* (pp. 162–173). Berlin: Springer.

Narula, S. C., & Ho, C. A. (1980). Degree-constrained minimum spanning trees. *Computers and Operations Research*, *7*, 239–249.

Palmer, C. C. (1994). *An approach to a problem in network design using genetic algorithms.* unpublished PhD thesis, Polytechnic University, Troy, NY.

Palmer, C. C., & Kershenbaum, A. (1994). Representing trees in genetic algorithms. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, Volume 1 (pp. 379–384). Piscataway, NJ: IEEE Service Center.

Papadimitriou, C. H., & Yannakakis, M. (1991). Optimization, approximation, and complexity classes. *J. Comput. System Sci.*, *43*, 425–440.

Prüfer, H. (1918). Neuer Beweis eines Satzes über Permutationen. *Archiv für Mathematik und Physik*, *27*, 742–744.

Raidl, G. R., & Julstrom, B. A. (2003). Edge-sets: An effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, *7*(3), 225–239.

Rothlauf, F. (2002). *Representations for genetic and evolutionary algorithms.* Number 104 in Studies on Fuzziness and Soft Computing. Berlin: Springer. 1st edition 2002. 2nd printing 2003.

Rothlauf, F., Gerstacker, J., & Heinzl, A. (2003). *On the optimal communication spanning tree problem* (Technical Report 15/2003). University of Mannheim.

Rothlauf, F., & Goldberg, D. E. (2003). Redundant representations in evolutionary computation. *Evolutionary Computation*, *11*(4), 381–415.

Rothlauf, F., Goldberg, D. E., & Heinzl, A. (2002). Network random keys – A tree network representation scheme for genetic and evolutionary algorithms. *Evolutionary Computation*, *10*(1), 75–97.