Making the Edge-Set Encoding Fly by Controlling the Bias of its Crossover Operator

Franz Rothlauf and Carsten Tzschoppe

Working Paper 12/2004
October 2004

Working Papers in Information Systems

University of Mannheim
Department of Business Administration and Information Systems
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`

Making the Edge-Set Encoding Fly by Controlling the Bias of its Crossover Operator

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

Carsten Tzschoppe
Dept. of Business Administration and Information Systems
University of Mannheim
D-68131 Mannheim/Germany
carsten.tzschoppe@gmx.de

October 8, 2004

Abstract

The edge-set encoding is a direct tree encoding which applies search operators directly to trees represented as sets of edges. There are two variants of crossover operators for the edge-set encoding: With heuristics that consider the weights of the edges, or without heuristics. Due to a strong bias of the heuristic crossover operator towards the minimum spanning tree (MST) a population of solutions converges quickly towards the MST and EAs using this operator show low performance when used for tree optimization problems where the optimal solution is not the MST. This paper presents a modified crossover operator (γ-TX) that allows us to control the bias towards the MST. The bias can be set arbitrarily between the strong bias of the heuristic crossover operator, or being unbiased. An investigation into the performance of EAs using the γ-TX for randomly created OCST problems of different types and OCST test instances from the literature present good results when setting the crossover-specific parameter γ properly. The presented results suggest that the original heuristic crossover operator of the edge-sets should be substituted by the modified γ-TX operator that allows us to control the bias towards the MST.

1 Introduction

A spanning tree T of an undirected graph $G(V, E)$ 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 [1], or the degree-constrained minimum spanning tree problem [2, 3]. When using evolutionary algorithms (EAs) for tree problems it is necessary to encode a tree such that the evolutionary search operators like crossover or mutation can be applied. There are two different possibilities for doing this. Indirect representations usually encode a tree (phenotype) as a list of strings (genotypes) and apply standard search operators to the genotypes. The phenotypes are constructed by an appropriate genotype-phenotype mapping (representation). In contrast, direct representations encode a tree as a set of edges and apply search operators directly to the set of edges. Therefore, no representation is necessary. Instead, tree-specific search operators must be developed, as standard search operators can not be used any more. Examples for direct encodings are the edge-set encoding [3], or the NetDir encoding [4, sec. 7.2]. [3] proposed two different variants of search operators for the edge-set encoding: Heuristic variants where the operators consider the weights of the edges, and non-heuristic versions. Results for the degree-constrained MST problem and the traveling salesman problem indicated a good performance of the heuristic variants [5, 3].

Representations resp. search operators can have a bias towards some solutions. A representation is biased if it is redundant and some phenotypes are over-represented. A search operator is biased if its iterative application results in a biased population that means not all possible phenotypes are represented with the same probability by the population. Consequently, heuristic search is biased if it pushes a population of solutions towards some solution even if no selection operator is used. Biased representations resp. operators do change the performance of evolutionary search. If the optimal solutions are similar to the solution where the representation resp. search operator is biased to, EA performance increases [4, section 3.1]. However, if there is a bias towards a solution that is not similar to the optimal solution, EA performance is low. [6, 7] examined the bias of the edge-set encoding and found a strong bias of the heuristic crossover operator of the edge-sets towards the MST. Therefore, problems where the optimal solution is the MST can be easily solved. However, as the bias of the heuristic crossover towards the MST is strong, EAs fail if the optimal solution is only slightly different from the MST.

This paper proposes a modified version of the heuristic crossover operator of the edge-set encoding that allows us to control the strength of the bias towards the MST. Therefore, the problems arising from the oversized bias of the heuristic crossover operator can be overcome and its bias can be adjusted according to the properties of the problem at hand. If it is known a-priori that optimal solutions of a problem are similar to the MST, a modest bias towards the MST allows EAs to solve the problem more efficiently. Experiments on the performance of the modified crossover operator are performed for the optimal communication spanning tree (OCST) problem. Results for random problems and problem instances from the literature show that by controlling the strength of the heuristic bias EA performance increases.

The paper is structured as follows. The following section describes the functionality of the edge-set encoding with and without heuristics and introduces the modified crossover operator. Section 3 investigates the bias of the crossover operators of the edge-set encoding and shows that the bias can be controlled when using the modified crossover operator. Its influence on EAs when solving OCST problems is examined in section 4. The paper ends with concluding remarks.

2 The Edge-Set Encoding

The edge-set encoding [3] is a direct representation for trees. Therefore, the search operators are applied directly to sets of edges. There are two different variants of search and initialization operators of the edge-set encoding: either with or without heuristics. When using operators with heuristics the weights of the edges are considered for the construction of the offspring. In the following paragraphs we briefly review the functionality of the initialization method and the crossover operator. We do not consider the mutation operator as [3] already proposed a version of the mutation operator that allows us to control its bias (compare [7]).

2.1 The Edge-Set Encoding without Heuristics

2.1.1 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. [3] have shown that this algorithm for creating random spanning trees, KruskalRST, has a small bias towards star-like trees.

procedure KruskalRST(V, E): //E: set of edges; V: set of vertices
$T \leftarrow \emptyset,\ A \leftarrow E$; //$T$: to be constructed spanning tree
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.

2.1.2 Recombination

The non-heuristic KruskalRST* crossover operator [3] includes in a first step all edges that are common to both parents T_1 and T_2 in the offspring T_{off}. Then, in a second step, KruskalRST is applied to $G_{cr} = (V, T_1 \cup T_2)$. KruskalRST* 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.

2.2 Heuristic Recombination Operators for the Edge-Set Encoding

The heuristic crossover operator presented in [3] is a modified version of KruskalRST* crossover. In a first step, 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'. This version of the heuristic crossover operator is denoted as 2-tournament-crossover (TX). [5] proposed two other variants of the heuristic crossover operator. They differ in the strategy of completing the offspring with the edges available in E':

- **Greedy crossover:** When using this strategy, the edge with the smallest weight is chosen from E'.
- **Inverse-weight-proportional crossover:** This strategy selects each edge from E' according to probabilities inversely proportional to the edges' weights.

[5] examined the performance of the different crossover variants for the traveling-salesperson problem and the degree-constraint MST problem. The results indicated that Greedy crossover shows good performance for simple and easy problem instances. For large problems TX crossover resulted in the best performance.

Due to the construction process all three crossover strategies have a strong bias towards the MST. The bias of the TX operator is already so strong that EAs are only able to find optimal solutions if they are very similar to the MST [7]. Problems where the optimal solutions are slightly different from the MST could no longer be solved by using the TX operator. According to the construction process the bias of the Greedy crossover is higher than the bias of the TX crossover. Therefore, Greedy crossover also results in low EA performance if the optimal solution is not the MST. The inverse-weight-proportional crossover introduces a bias to the MST similar to the TX crossover. However, the bias can not be controlled in a systematic way but depends on the specific weights of the edges.

We want to propose a modified version of the heuristic TX operator. The modification is only small but allows us to control the bias towards the MST. In the new crossover variant (denoted as γ-TX crossover) the tournament of size two that chooses one edge from E' is not always performed but only with the probability γ. Therefore, for $\gamma = 0$ an edge is randomly chosen from E' and we see the same behavior as KruskalRST*. For $\gamma = 1$ all edges are chosen by a tournament of size 2 and we get the same behavior as TX crossover. The bias of γ-TX towards the MST can be set arbitrarily small with $\gamma \to 0$.

3 Bias of the Crossover Operators for Edge-Sets

We investigate the bias of the TX and γ-TX operator for randomly created trees with $n = 10$ and $n = 16$ nodes. To every edge (i, j) a non-negative weight

w_{ij} is associated. We want to consider two different possibilities for the weights w_{ij}:

- **Random weights:** The real-valued weights w_{ij} are generated randomly and are uniformly distributed in $]0, 100]$.
- **Euclidean weights:** The nodes are randomly placed on a 1000x1000 grid. The weights w_{ij} between nodes i and j are the Euclidean distances between nodes i and j.

As the weights w_{ij} are randomly created and $w_{ij} \neq w_{kl}$, $\forall i \neq k, j \neq l$, there is a unique MST for every problem instance. T is the MST 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 an edge from u to v exists in T_i and 0 if it does not exist in T_i.

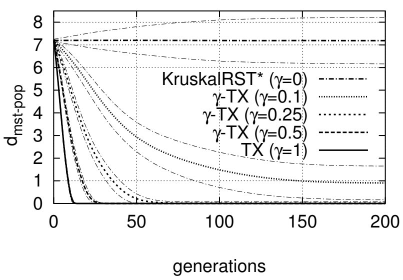
For the experiments we randomly generate an initial population of 500 individuals using the non-heuristic KruskalRST initialization and apply the crossover operators iteratively. As no selection operator is used, no selection pressure pushes the population to high-quality solutions. An operator is unbiased if the statistical properties of the population do not change by applying crossover alone. In the 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 MST. If $d_{mst-pop}$ decreases, the crossover operator is biased towards the MST. If $d_{mst-pop}$ remains constant, the crossover operator is unbiased and no MST-like solutions are overrepresented.

We performed this experiment on 500 randomly generated 10 and 16 node problem instances with random, resp. Euclidean weights w_{ij}. For every problem instance we performed 50 runs. In each run, the crossover operator was applied 200 generations. Fig. 1 shows the mean and the standard deviation of the distance $d_{mst-pop}$ over the number of generations. The plots compare the non-heuristic KruskalRST* crossover with the heuristic TX and γ-TX operator (no selection is used).

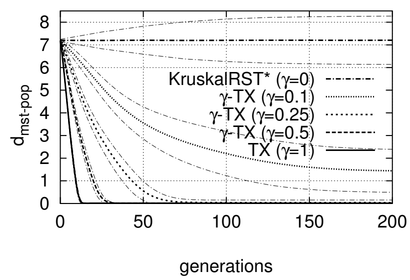
The results show that the non-heuristic KruskalRST* operator is unbiased. In contrast, the heuristic TX operator shows a strong bias towards the MST and a population converges to the MST after a few generations. When using the γ-TX crossover the bias towards the MST can be controlled. With lower γ the bias gets smaller and for $\gamma = 0$ we get the same results as for KruskalRST*.

4 Performance of the γ-TX Crossover for OCST Problems

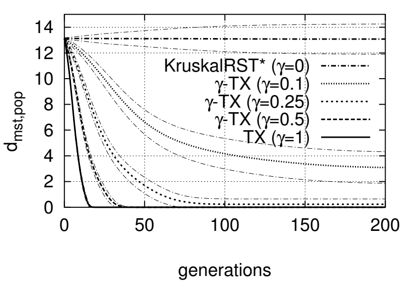
This section investigates how the performance of different crossover variants of the edge-set encoding depends on the properties of the optimal solutions. We perform the experiments for the optimal communication spanning tree (OCST) problem as all trees are feasible solutions and there are no additional constraints.



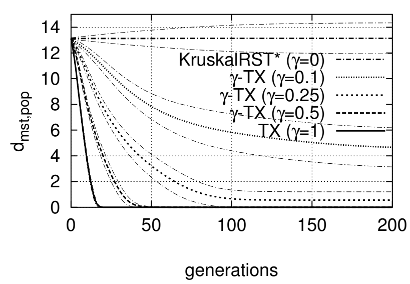
(a) 10 node, random weights



(b) 10 node, Euclidean weights



(c) 16 node, random weights



(d) 16 node, Euclidean weights

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 MST over the number of generations when only using crossover (no selection pressure). The results show that the bias of the γ-TX crossover can be controlled and lies between the strong bias of TX crossover ($\gamma = 1$) and the no-bias of the KruskalRST* crossover ($\gamma = 0$).

4.1 Optimal Solutions for Randomly Created OCST Problems

The OCST problem was first introduced in [1] and is $\mathcal{MAX}\ \mathcal{NP}$-hard [8]. The problem seeks a spanning tree that connects all given nodes and satisfies their communication requirements for a minimum total cost. The problem can be defined as follows: Let $G = (V, E)$ be a complete undirected graph with $n = |V|$ nodes and $m = |E|$ edges. 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. The OCST problem becomes the MST problem if there are no communication requirements r_{ij} and $c(T) = \sum_{(i,j)\in T} w_{ij}$.

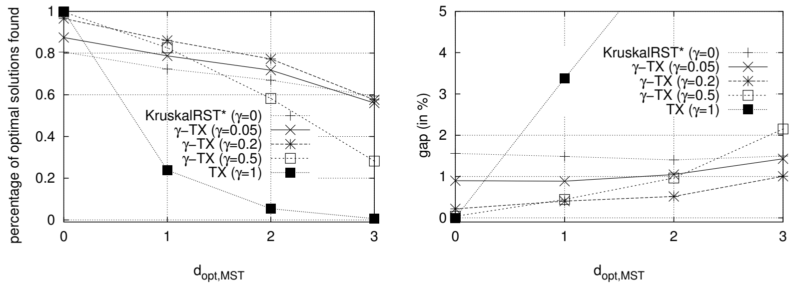
It was shown in [9] that on average optimal solutions for OCST problems are similar to the MST, that means the average distance $d_{opt,MST}$ between the optimal solution and the MST is significantly lower than the average distance $d_{rand,MST}$ between a randomly created tree and the MST. Therefore, as the optimal solutions of OCST problems are biased towards the MST, representations as well as operators that are biased to the MST are expected to solve the OCST problem efficiently.

To investigate how the performance of EAs using different crossover variants of edge-sets depend on the structure of the optimal solution, an optimal or near-optimal solution for the OCST problem must be determined. We identified optimal (or near-optimal) solutions for the OCST problem by an EA whose population size N is doubled in every iteration until the same solutions are found in subsequent iterations. Details of the experimental setting for finding optimal solutions for OCST problems can be found in [9].

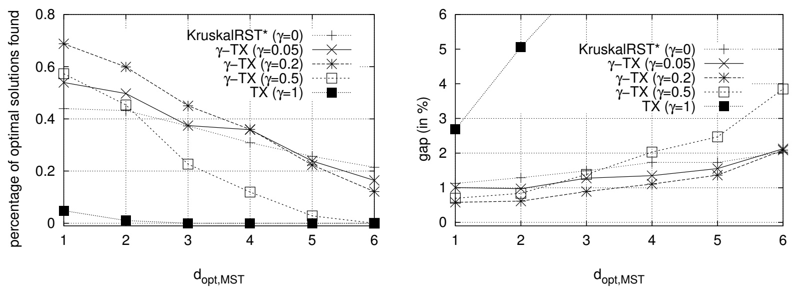
4.2 Edge-Set Crossover for Randomly Created OCST Problems

This section investigates for randomly created OCST problems how the performance of EAs using different variants of the crossover operator depends on the distance $d_{opt,MST}$ between the optimal solution and the MST.

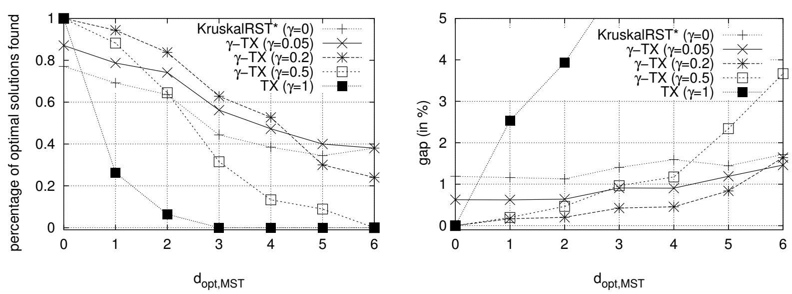
We randomly generated 500 problem instances with 10 and 16 nodes using either random or Euclidean distance weights. The demands r_{ij} are chosen randomly and are uniformly distributed in]0,. . . ,100]. Then, we determine the optimal solutions using the experimental setting described in [9]. For comparing the performance of the different crossover variants (KruskalRST*, TX, and γ-TX) we use a simple generational EA with no mutation and tournament selection without replacement of size two. The population size N is chosen with respect to the performance of KruskalRST*. The aim is to find the optimal solution with a probability of about 25-75 %. Therefore, we choose for the 10 node problems a population size $N = 100$ and for the 16 node problems $N = 250$. Each run is stopped after the population is fully converged or the



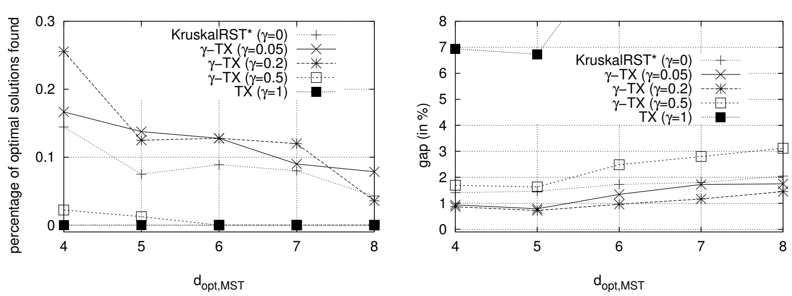
(a) 10 node, random weights



(b) 10 node, Euclidean weights



(c) 16 node, random weights



(d) 16 node, Euclidean weights

Figure 2: The plots show the percentage of optimal solutions that can be found (left) and the gap between the cost of the best found solution and the optimal solution (right) over $d_{opt,MST}$ for different crossover operators and different types of OCST problems. The plots show that with increasing $d_{opt,MST}$, EAs using the TX operators fail due to their strong bias. When using the γ-TX operator with a low bias, EA performance is high. With increasing γ the bias towards the MST becomes stronger and problems with larger $d_{opt,MST}$ can no longer be solved.

number of generations exceeds 200. 50 runs are performed for each of the 500 problem instances.

The results of our experiments are presented in Fig. 2. It shows the percentage of EA runs that find the optimal solutions (left) and the gap $\frac{c(T_{found})-c(T_{opt})}{c(T_{opt})}$ between the cost of the optimal solution T_{opt} and the cost of the best found solution T_{found} (right) at the end of a run over the distance $d_{opt,MST}$ between the optimal solution and the MST. Results are plotted for KruskalRST*, different variants of γ-TX, and TX. The initial population was generated using the non-heuristic initialization from section 2.1. We only show results for those $d_{opt,MST}$ with more than 10 problem instances (out of 500).

The results reveal that with increasing $d_{opt,MST}$ the performance of EAs is reduced. The decrease in performance is emphasized with larger γ. When using a crossover operator with a strong bias like TX or γ-TX with $\gamma = 0.5$ EA performance is high if and only if $d_{opt,MST} \approx 0$; with larger $d_{opt,MST}$ EA performance drops rapidly. The strong bias pushes the population towards the MST and makes it difficult to find the optimal solution. In contrast, when using the γ-TX operator with a low γ ($\gamma = 0.05$ or $\gamma = 0.2$) the bias towards the MST is small and reasonable and EAs perform better or equal than when using the non-heuristic version. These results are confirmed when examining the gap $\frac{c(T_{found})-c(T_{opt})}{c(T_{opt})}$. With increasing bias and increasing $d_{opt,MST}$, the quality of the found solutions decreases.

In summary, using a strong bias towards the MST results in high EA performance for $d_{opt,MST} \approx 0$ but low performance elsewhere. With lower γ, problems with larger $d_{opt,MST}$ can be solved. EAs using the γ-TX operator with a low bias towards the MST ($\gamma \approx 0.05 - 0.2$ for 10 nodes and $\gamma \approx 0.05$ for 16 node problems) outperform the non-heuristic KruskalRST* crossover for low $d_{opt,MST}$ and also show good results for larger $d_{opt,MST}$.

4.3 Edge-Set Crossover for Test Instances from the Literature

Test instances for the OCST problem have been proposed by [10, 11, 12]. Details of the test instances and an analysis of their properties can be found in [9]. The following paragraphs examine the performance of EAs using the different crossover variants for these test instances.

Table 1 lists the properties of the optimal solutions for the test instances. It shows the number of nodes n, the distance $d_{opt,MST}$, and the cost $c(T_{opt})$ of the optimal solution. In the instance berry35u, all distances are uniform ($w_{ij} = 1$), so all spanning trees are minimal. For all test instances, $d_{opt,MST}$ is smaller than the average distance of a randomly created solution towards the MST (compare [9]). Therefore, all test problems are biased towards the MSTs.

For the experiments the same generational EA with population size N as in the previous section is used. The table presents the mean μ and the standard deviation σ of the cost of the best solution found at the end of the runs averaged over 50 runs for each problem instance. The results show that the heuristic TX crossover ($\gamma = 1$) is only able to find the optimal solution if $d_{opt,MST} = 0$. Otherwise, the performance of EAs using it is low. In contrast, the performance of EAs using the non-heuristic and unbiased KrukalRST* is high and high

Table 1: Performance of EA using different crossover operators for OCST test problems from the literature

problem		opt solution			KruskalRST*		γ-TX ($\gamma = 0.05$)		γ-TX ($\gamma = 0.2$)		TX ($\gamma = 1$)	
instance	n	$d_{opt,MST}$	$c(T_{opt})$	N	μ	σ	μ	σ	μ	σ	μ	σ
palmer6	6	1	693,180	50	698,200	8,447	696,301	5,833	698,217	8,524	706,784	6,438
palmer12	12	7	3,428,509	60	3,589,154	92,885	3,582,433	91,529	3,534,935	71,125	3,707,947	56,691
palmer24	24	12	1,086,656	800	1,088,231	915	1,088,007	665	1,088,615	695	1,873,835	36,453
raidl10	10	3	53,674	60	57,046	5,266	55,275	3,481	55,077	2,261	57,200	927
raidl20	20	4	157,570	400	159,714	4,038	159,943	3,984	157,922	1,426	164,811	2,671
berry6	6	0	534	50	539	13	534	3	534	0	534	0
berry35u	35	-	16,273	800	16,621	173	16,622	180	16,604	190	16,577	187
berry35	35	0	16,915	800	17,263	381	16,975	138	16,915	0	16,915	0

quality solutions can be found. When using the modified γ-TX operator with a low γ ($\gamma = 0.05$ or $\gamma = 0.2$) the performance of EAs can be increased in comparison to KruskalRST* as the optimal solutions for all problem instances are biased towards the MST. EAs using the γ-TX with low γ (e. g. $\gamma = 0.05$) always find solutions of similar or higher quality. Only for berry35u can no better solutions be found as all spanning trees are minimal and therefore a low bias towards the MST does not increase EA performance.

5 Summary and Conclusions

This work proposed a new variant (γ-TX) of the heuristic crossover (TX) operator of the edge-set encoding. When using the standard TX operator an offspring tree is created from two parents by inserting all edges that are common in both parents into the offspring. The offspring is completed by parental edges chosen by a tournament of size two. Edges with lower weight are preferred. In contrast to the standard TX operator, the γ-TX operator only performs a tournament with probability γ and otherwise inserts a random edge from one of the parents.

Due to its construction rule the TX operator shows a strong bias towards the minimum spanning tree (MST). Using it for the optimal communication spanning tree (OCST) problem allows EAs only to solve the problem if the optimal solution is the MST. If the optimal solution is slightly different from the MST, EAs fail. The γ-TX operator allows us to control the bias which can be set arbitrarily (according to γ) between the strong bias of the TX operator ($\gamma = 1$) and no-bias ($\gamma = 0$). Therefore, the problems of the TX operator with the strong bias towards the MST can be overcome while still allowing the crossover operator to be slightly biased towards the MST. The experimental results for random OCST problem instances and problem instances from the literature show that EAs using the γ-TX operator with a proper setting of γ show good performance.

The problems of the TX operator of the edge-set encoding emphasize the difficulties of a proper design of representations and operators. If it is known a priori that the optimal solutions for some problems are biased towards some solutions this bias can be exploited by developing representations resp. operators that are biased in the same direction. Then, with a proper bias, problems can be solved more efficiently than when using non-biased encodings. However if the bias is too great, EAs fail if the optimal solution is only slightly different from the solution the representation resp. operator is biased to. Therefore, biased representations should be used with great care and only if there is some a priori knowledge regarding the properties of the optimal solutions. Otherwise, if no a priori knowledge exists non-biased representations should be preferred.

References

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

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

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

[4] Rothlauf, F.: Representations for Genetic and Evolutionary Algorithms. 1 edn. Number 104 in Studies on Fuzziness and Soft Computing. Springer, Heidelberg (2002)

[5] Julstrom, B., Raidl, G.: Weight-biased edge-crossover in evolutionary algorithms for two graph problems. In et al., G.L., ed.: Proceedings of the 16th ACM Symposium on Applied Computing, ACM Press (2001) 321–326

[6] Tzschoppe, C., Rothlauf, F., Pesch, H.J.: The edge-set encoding revisited: On the bias of a direct representation for trees. In Deb, Kalyanmoy et al., ed.: Proceedings of the Genetic and Evolutionary Computation Conference 2004, Heidelberg, Springer (2004) 1174–1185

[7] Rothlauf, F., Tzschoppe, C.: On the bias and performance of the edge-set encoding. Technical Report 2004/11, Department of Information Systems, University of Mannheim (2004)

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

[9] Rothlauf, F., Gerstacker, J., Heinzl, A.: On the optimal communication spanning tree problem. Technical Report 15/2003, Department of Information Systems, University of Mannheim (2003)

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

[11] Berry, L.T.M., Murtagh, B.A., McMahon, G.: 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, Pretoria, South Africa (1995) 361–370

[12] Raidl, G.R.: Various instances of optimal communication spanning tree problems. personal communciation (2001)