On the Optimal Communication Spanning Tree Problem

Franz Rothlauf, Jürgen Gerstacker, and Armin Heinzl

Working Paper 10/2003
May 2003

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`

On the Optimal Communication Spanning Tree Problem

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

Jüergen Gerstacker
Dept. of Information Systems
University of Bayreuth
D-95440 Bayreuth/Germany
gerstacker@uni-bayreuth.de

Armin Heinzl
Dept. of Information Systems 1
University of Mannheim
D-68131 Mannheim/Germany
heinzl@bwl.uni-mannheim.de

May 3, 2003

Abstract

This paper presents an investigation into the properties of the optimal communication spanning tree (OCST) problem. The OCST problem finds a spanning tree that connects all nodes and satisfies their communication requirements for a minimum total cost. The paper compares the properties of randomly created solutions to the best solutions that are found using an evolutionary algorithm framework. The results show that on average the distance between the optimal solution and the minimum spanning tree (MST) that is calculated according to the distance weights is significantly smaller than the distance between a randomly created solution and the MST. This means, optimal solutions for the OCST problem are biased towards the MST defined on the distance weights alone. Consequently, the performance of optimization methods for the OCST problem can be increased if the search is biased towards MST-like solutions.

1 Introduction

The optimal communication spanning tree (OCST) problem (Hu, 1974) finds a spanning tree that connects all given nodes and satisfies their communication requirements for a minimum total cost. The number and positions of the network nodes are given a priori and the cost of the network is determined by the cost of the links. Like other constrained spanning tree problems, the OCST problem is NP-hard (Garey & Johnson, 1979).

This paper presents an investigation into the properties of optimal solutions for OCST problems. It examines the properties of existing OCST problem instances from the literature, as well as analyzing OCST problems in general. We perform experiments using Euclidean and randomly chosen distance weights. The results show that the distance between optimal solutions for the OCST problem and the minimum spanning tree (MST) that can be calculated using the given distance weights is smaller than the distance between a randomly created solution and the MST. Therefore, optimal solutions for OCST problems are biased towards the MST. As a consequence, search methods that consider this problem-specific knowledge are expected to show good performance for the OCST problem.

The paper is structured as follows. In the following section we give a short description of the OCST problem and examine existing problem instances from the literature. We analyze the properties of randomly created trees and compare them to the properties of the best known solutions. Consequently, in section 3 we perform a statistical analysis of OCST problems in general. We generate random problem instances and compare the properties of randomly created trees to the properties of the optimal solutions. Section 3.3 discusses the impact of the results on the design

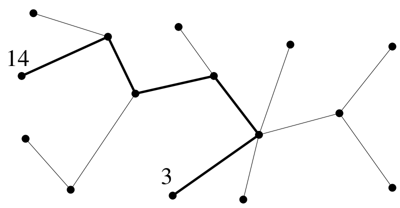

Figure 1: A communication spanning tree on 15 nodes where the path connecting nodes 3 and 14 is emphasized.

of heuristic optimization methods especially evolutionary algorithms. In section 4 we present some directions of future work. The paper ends with concluding remarks.

2 The Optimal Communication Spanning Tree Problem

In the following we define the OCST problem and analyze existing problem instances from the literature.

2.1 Problem Description

The design of optimal communication and transportation networks which satisfy a given set of requirements has been studied extensively in the literature. Many different variants, with or without additional constraints have been examined, and either exact solutions or heuristics have been given (Kershenbaum, 1993; Cahn, 1998). Relevant constrained minimum spanning tree problems are for example the optimal communication spanning tree problem, the degree-constrained minimum spanning tree problem, the minimum steiner tree problem, or the capacitated minimum spanning tree problem.

For the optimal communication spanning tree (OCST) problem (Hu, 1974), a tree that connects all given nodes and satisfies their communication requirements for a minimum total cost has to be found. The number and positions of the network nodes are given a priori and the cost of the network is determined by the cost of the links. A link's flow is the sum of the communication demands between all pairs of nodes communicating either directly, or indirectly, over the link. Figure 1 shows a communication spanning tree on 15 nodes and emphasizes the path connecting nodes 3 and 14. The cost for each link is not fixed a priori but depends on its length and capacity. A link's capacity must satisfy the flow over this link, which depends on the entire tree structure.

The OCST problem can be defined formally as follows. An undirected graph is denoted as $G = (V, E)$. $n = |V|$ denotes the number of nodes and $m = |E|$ denotes the number of edges of the graph. There are communication or transportation demands between the n different nodes. The demands are specified by an $n \times n$ *demand matrix* $R = (r_{ij})$, where r_{ij} is the amount of traffic required between location v_i and v_j. An $n \times n$ *distance matrix* $W = w_{ij}$ determines the distance weights associated with each pair of sites. A tree $T = (V, F)$ where $F \subseteq E$ and $|F| = |V| - 1$ is called a *spanning tree* of G if it connects all the nodes. The weight $w(T)$ of the spanning tree is the weighted sum over all pairs of vertices of the cost of the path between the pair in T. In general,

$$w(T) = \sum_{i,j \in V} f\left(w_{ij}, b_{ij}\right),$$

where $B = b_{ij}$ denotes the traffic flowing directly and indirectly between the nodes i and j. It is calculated according to W and the structure of T. T is the optimal communication spanning tree if $w(T) \leq w(T')$ for all other spanning trees T'.

In many problem instances the cost of a link is calculated as the product of the distance weight times the overall traffic running over the link. Therefore, $f = w_{ij} * b_{ij}$. The OCST problem becomes the minimum spanning tree (MST) problem if $f = w_{ij}$. Then, T is the minimum spanning tree if $w(T) \leq w(T')$ for all other spanning trees T', where $w(T) = \sum_{i,j \in V} w_{ij}$.

Cayley's formula identified the number of spanning trees on n nodes as n^{n-2} (Cayley, 1889). Furthermore, there are n different stars on a graph of n nodes. A distance $d_{ij} \in \{0, 1, \ldots, n-2\}$ between two spanning trees T_i and T_j can be defined as

$$d_{ij} = \frac{1}{2} \sum_{u,v \in V} |l^i_{uv} - l^j_{uv}|.$$

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. The number of links that two trees T_i and T_j have in common can be calculated as $n - 1 - d_{ij}$.

Like other constrained spanning tree problems, the OCST problem is NP-hard (Garey & Johnson, 1979, p. 207). However, previous work has shown that the problem can be solved with $O(\log^2(|V|)$ if the distance matrix satisfies the triangle inequality (Peleg & Reshef, 1998). A large number of evolutionary algorithms using different types of search operators and representations have been proposed for solving the OCST problem (Davis, Orvosh, Cox, & Qiu, 1993; Berry, Murtagh, & Sugden, 1994; Kim & Gen, 1999; Davis, Orvosh, Cox, & Qiu, 1993; Tang, Man, & Ko, 1997; Sinclair, 1995; Berry, Murtagh, & Sugden, 1994; Kim & Gen, 1999; Li & Bouchebaba, 1999; Raidl & Julstrom, 2000; Gottlieb, Julstrom, Raidl, & Rothlauf, 2001; Julstrom, 2001; Chou, Premkumar, & Chu, 2001; Rothlauf, Goldberg, & Heinzl, 2002).

2.2 Analysis of Existing Problem Instances

Test instances for the OCST problem have been proposed in the literature by Palmer (1994), Berry, Murtagh, and McMahon (1995), Rothlauf, Goldberg, and Heinzl (2002), and Raidl (2001). In the following we analyze the properties of these test instances.

Palmer described in his thesis OCST problems with 6 (palm6), 12 (palm12), 24 (palm24), 47, and 98 nodes. The inter-node traffic demands are inversely proportional to the distances between the nodes. The nodes correspond to cities in the United States and the distances between the nodes are obtained from a tariff database. For the exact distance and requirement matrix for the 6, 12 and 24 node problem the reader is referred to Palmer (1994).

Berry, Murtagh, and McMahon presented three different instances of the OCST problem. A six node (berry6) and two 35 node problems (berry35 and berry35u) have been proposed. For berry35u the distance weights $w_{ij} = 1$. It is known that the cost of the optimal solution for berry6 is 534 and for berry35 is 16 915. The distance matrix and the traffic demands can also be found at `http://www.cse.rmit.edu.au/~rdslw/research.html`. Both the problems from Palmer and Berry, Murtagh, and McMahon, have also been investigated by Li and Bouchebaba (1999).

Rothlauf (2002) presented four OCST problems that are derived from a real-world 26-node problem from a company with locations all over Germany. For fulfilling the demands between the nodes, different line types with only discrete capacities and costs are available. The costs for installing a link consists of a fixed and length dependent share. Both depend on the capacity of the link. The cost are based on the tariffs of the German Telecom from 1996. The distances between the nodes (cities) are calculated using Euclidean distances. For an exact description on how the cost of a link depends on its length and its capacity the reader is referred to Rothlauf (2002).

Finally, Raidl (2001) proposed several test instances ranging from 10 to 100 nodes. The distance weights and the traffic demands have been generated randomly. They are uniformly distributed in the interval $[0, 100]$. The distance matrix and traffic demands can be obtained directly from the author[1].

In the test instances from Palmer, Berry et al., and Raidl, the weight $w(T)$ of a tree is defined as

$$w(T) = \sum_{i,j \in V} w_{ij} * b_{ij},$$

where w_{ij} is the distance weight between node i and j and b_{ij} is the sum of direct and indirect traffic traversing the link between i and j. In the test instances from Rothlauf, there are discrete link capacities available and the cost of a link $w(E)$ is calculated according to a tariff database. A detailed description of all test problems can be found at Rothlauf (2002), which is also available at `http://www.bwl.uni-mannheim.de/wifo1/de/gea_book.htm`.

In the following, we perform an investigation into specific properties of these test problems. The goal is to gain additional information about the problems as well as about the optimal solutions. To gain an idea about the properties of randomly generated solutions, we randomly generate 10 000 solutions (trees) for each test problem. As it is difficult to generate random trees in an unbiased manner (Raidl & Julstrom, 2003), we encode the trees using the Prüfer number representation (Prüfer, 1918). Generating random individuals by generating random Prüfer numbers allows us to create unbiased solutions. The Prüfer number representation is a one-to-one mapping between spanning trees on n nodes and strings of length $n-2$ using an alphabet of cardinality n. The use of this representation ensures that the probability of generating random trees is uniformly distributed and no trees are favored.

Table 1 lists the properties of randomly created solutions. It shows the mean μ and the standard deviation σ of the distance $d_{mst,rand}$ between a randomly generated solution and the minimum spanning tree. As described in the previous subsection, the MST is calculated using only the distance weights w_{ij}. For the problem instance berry35u we are not able to calculate $d_{mst,rand}$, as all distances are uniform ($w_{ij} = 1$). Furthermore, we calculated the distances between a randomly created solution and the n different stars, $d_{star,best}$. The minimum distance between a randomly created solution and one of the n stars, measures the similarity of a randomly created solution towards a star. Table 1 shows the mean μ and standard deviation σ of $\min(d_{star,rand})$ that denotes the minimal distance between a randomly generated solution and a star.

Consequently, we calculated for the optimal or best known solution of the considered test problems the distance towards the MST, $d_{mst,best}$, and the minimum distance towards one of the n stars, $\min(d_{star,best})$. The results, including the cost of the optimal solution, are shown in Table 2. Comparing $d_{mst,best}$ with $d_{mst,rand}$ reveals that for all test instances $d_{mst,best} < d_{mst,rand}$. This means that for all test instances the best solution is much more similar to the MST in comparison to a randomly generated solution. Therefore, the optimal solution is biased towards the MST. Comparing $\min(d_{star,best})$ and $\min(d_{star,rand})$ does not reveal any significant differences. Randomly created solutions have about the same minimum distance towards a star in comparison to the optimal solution. Therefore, the optimal solution is not significantly biased towards a star network.

3 Statistical Analysis of the OCST problem

In the previous section we have seen that the optimal (or best known) solutions to the considered test problems are biased towards the MST. Consequently, the following section performs a more general, statistical, analysis of OCST problems. We randomly generate a large number of different OCST

[1]e-mail: raidl@apm.tuwien.ac.at

problem instance	nodes	$d_{mst,rand}$		$\min(d_{star,rand})$	
		μ	σ	μ	σ
palmer6	6	3.36	0.91	2.04	0.61
palmer12	12	9.17	1.17	7.22	0.75
palmer24	24	21.05	1.30	18.50	0.80
raidl10	10	7.20	1.10	5.42	0.70
raidl20	20	17.07	1.27	14.69	0.77
berry6	6	3.51	0.83	2.03	0.61
berry35u	35	-	-	29.19	0.83
berry35	35	32.05	1.32	29.16	0.83
rothlauf2	15	12.08	1.08	9.99	0.772
rothlauf1					
rothlauf3	16	13.07	1.24	10.89	0.80
rothlauf4					

Table 1: Properties of randomly created solutions for the test instances

problem	$d_{mst,best}$	$\min(d_{star,best})$	cost
palmer6	1	2	693 180
palmer12	5	7	3 428 509
palmer24	12	17	1 086 656
raidl10	3	4	53 674
raidl20	4	14	157 570
berry6	0	2	534
berry35u	-	28	16 273
berry35	1	30	16 915
rothlauf2	4	8	58 619
rothlauf1	7	9	60 883
rothlauf3	6	9	28 451
rothlauf4	9	7	112 938

Table 2: Properties of the optimal solutions for the test instances.

problem instances and compare the properties of randomly generated solutions to the properties of the best solution that is found using an evolutionary algorithm approach.

3.1 Randomly Generated Solutions for the OCST Problem

We created for each problem size 100 random problem instances. The demands r_{ij} between two nodes i and j are generated randomly and are uniformly distributed in the interval $[0, 100]$. For the distances w_{ij} there are two possibilities:

- Random distance: The distances w_{ij} are randomly generated and are uniformly distributed in the interval $[0, 100]$.
- Euclidean distance: The nodes are randomly placed on a 1000x1000 2-dimensional plane. The distance w_{ij} between nodes i and j is the Euclidean distance between the two nodes.

The weight $w(T)$ of a tree is defined as

$$w(T) = \sum_{i,j \in V} w_{ij} * b_{ij}.$$

For each problem size n (number of nodes) 100 problem instances with randomly chosen positions of the nodes, either random or Euclidean distances, and random demands are created. For each of the 100 problem instance we generated 10 000 random solutions.

	$d_{mst,rand}$		$\min(d_{star,rand})$	
nodes	μ	σ	μ	σ
8	5.25	1.04	3.74	0.62
10	7.19	1.11	5.50	0.67
12	9.17	1.16	7.31	0.72
14	11.14	1.20	9.15	0.75
16	13.13	1.22	11.00	0.76
18	15.11	1.24	12.88	0.77
20	17.10	1.263	14.78	0.77
22	19.09	1.27	16.69	0.77
24	21.09	1.29	18.60	0.80
26	23.06	1.30	20.44	0.80

Table 3: Properties of randomly created solutions for the OCST problem

Table 3 presents the mean μ and standard deviation σ of $d_{mst,rand}$ and $\min(d_{star,rand})$. As we get the same results for using random distances and Euclidean distances we neglect the distances used. $d_{mst,rand}$ and $\min(d_{star,rand})$ are defined as described in the previous section. It can be seen that both the distance $d_{mst,rand}$ of a randomly generated solution towards the MST and the minimum distance $\min(d_{star,rand})$ towards a star, increase approximately linearly with the number of nodes n.

3.2 Optimal Solutions for the OCST Problem

In the previous section we examined the properties of randomly created solutions for the OCST problem. In the following section, we analyze the properties of the optimal solutions for randomly generated test instances and compare their properties to randomly created solutions. The optimal solutions are determined using an evolutionary algorithm framework.

As there are no exact optimization algorithms available that can solve even small instances of the OCST problem in a reasonable time, we implemented an evolutionary algorithm (EA) framework for determining the optimal solution. Although EAs are heuristic search methods that can not guarantee that the optimal solution is really found, we design an EA framework such that we can assume that the found solution is the optimal solution. To find the optimal solution for an OCST problem we use a classical canonical GA (Goldberg, 1989) with crossover as the main search operator and some background mutation.

We know from previous work (Harik, Cantú-Paz, Goldberg, & Miller, 1997) that the probability of GA failure α goes with $O(\exp(-N))$, where N is the used population size for the GA. Therefore, GA performance increases with increasing N. Consequently, 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 w_0 that is found during the n_{iter} runs. In a next round we double the population size and apply again a GA n_{iter} times with a population size of $N_1 = 2 * N_0$. T_1^{best} denotes the best solution with cost w_1 that can be found in the second round. We continue this iteration and double the population size $N_i = 2N_{i-1}$ until $w_i = w_{i-1}$. 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 use a standard GA with traditional parameter settings. The GA uses one-point crossover and tournament selection without replacement. The size of the tournament is three. The crossover probability is set to $p_{cross} = 0.7$ and the mutation probability is set to $p_{mut} = 0.02$. To encode trees we used the network random key representation proposed in Rothlauf, Goldberg, and Heinzl (2002). We have chosen this representation as it ensures good EA performance and as is unbiased that means all possible trees are represented uniformly and no trees are overrepresented by the representation.

In our experiments we applied the EA framework for each problem size n to the same 100 problem instances that we already examined in subsection 3.1. In contrast to subsection 3.1 where we characterized only randomly created solutions, we investigate in the following the properties of the corresponding optimal solutions. For the EA framework we started with $N_0 = 100$ and set $n_{iter} = 20$. The computational effort for the experiments is high. For calculating the optimal solutions of all 100 problem instances for a problem size of $n = 26$ using the proposed EA framework, we spent some 100 hours of computing time on a P4 with 2000 MHz.

	$d_{mst,opt}$		$\min(d_{star,opt})$		N_i	w_i	
nodes	μ	σ	μ	σ	μ	μ	σ
8	2.00	1.03	3.12	0.86	200	874 382	173 423
10	3.15	1.39	4.37	0.93	200	1 476 160	281 993
12	4.39	1.34	5.87	1.03	200	2 231 256	330 089
14	5.63	1.47	7.43	1.28	200	3 071 787	418 605
16	7.21	1.58	9.15	1.30	202	4 167 278	473 867
18	7.70	1.64	11.23	1.12	412	5 171 294	520 323
20	9.51	1.63	12.76	1.19	466	6 556 061	728 244
22	11.49	1.84	13.99	1.51	1154	8 131 648	869 736
24	12.38	1.89	16.62	0.96	2584	9 868 842	820 129
26	15.11	2.10	17.76	1.34	4673	11 615 434	858 822

Table 4: Properties of the optimal solutions using Euclidean distances

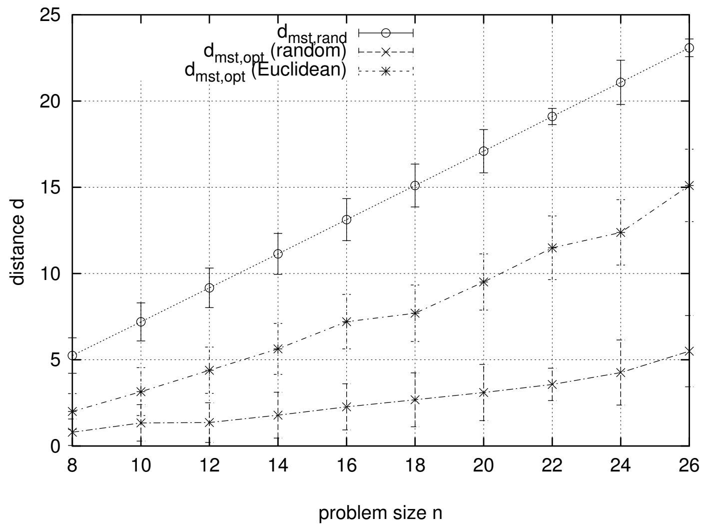

Figure 2: We show how $d_{mst,rand}$ and $d_{mst,opt}$ using random distances and Euclidean distances depend on the problem size n. The error bars indicate the standard deviation. The distances towards the MST increase linearly. As $d_{mst,opt} < d_{mst,rand}$, optimal solutions are biased towards the MST.

	$d_{mst,opt}$		$\min(d_{star,opt})$		N_i	w_i	
nodes	μ	σ	μ	σ	μ	μ	σ
8	0.80	0.77	3.45	0.74	200	47 693	18 986
10	1.34	1.06	4.92	0.86	200	70 311	23 518
12	1.36	1.15	6.79	0.80	200	89 787	35 662
14	1.79	1.33	8.54	0.93	200	108 532	32 522
16	2.27	1.34	10.41	0.82	400	136 097	36 606
18	2.68	1.56	12.03	1.00	800	157 491	42 776
20	3.10	1.63	14.12	1.01	800	184 315	47 604
22	3.57	1.74	15.71	0.83	857	205 240	43 749
24	4.26	1.88	17.81	0.98	1508	234 676	58 950
26	5.50	2.07	19.25	1.13	6250	265 768	58 088

Table 5: Properties of the optimal solutions using random distances

Table 4 and 5 present the properties of the best solutions that have been found for the 100 problem instances of each problem size n using the proposed EA framework. We distinguish between Euclidean (Table 4) and random distances (Table 5). The tables show for different problem sizes n the mean μ and standard deviation σ of the distance between the optimal solution and the MST, $d_{mst,opt}$, the minimum distance between the optimal solution and a star, $\min(d_{star,opt})$, and the cost w_i of the best solution T_i^{best}. Furthermore, they show the average population size N_i in the last GA round i.

The Tables reveal a significant difference for $d_{mst,opt}$ between using Euclidean distances and random distances. When using random distances, the optimal solutions are more similar to the MST than when using Euclidean distances. Comparing the properties of the optimal solutions to the properties of randomly created solutions listed in Table 3 reveals that $d_{mst,opt} \ll d_{mst,rand}$. This means, the optimal solutions to OCST problems are strongly biased towards the MST. Focusing on the minimum distances towards a star shows that there is a small bias towards stars especially if we use Euclidean distances. However, the effect is only weak and can be neglected in comparison to the strong bias towards the MST.

Figure 2 summarizes the results from Tables 3, 4, and 5 and plots $d_{mst,rand}$, $d_{mst,opt}$ using random distances, and $d_{mst,opt}$ using Euclidean distances over the problem size n. It can be seen that all distances towards the MST increase linearly with n. Both, the optimal solutions for OCST problems using Euclidean and random distances are strongly biased towards the MST.

3.3 Discussion

The insight that the optimal solutions to OCST problems are biased towards the MST can be beneficially used for the design of optimization methods. Optimization methods can perform better if they consider problem-specific knowledge. When using EAs for solving OCST problems, the insight that the optimal solution is more similar to the MST than a randomly created solution can be considered for the design of the search method. In the following we outline three possible approaches that are relevant to the design of evolutionary approaches:

- Overrepresent solutions similar to the MST in the initial population.
- Design search operators that favor trees similar to the MST.
- Increase the fitness of solutions similar to the MST.

In the following we briefly discuss these three possibilities. When using redundant representations, the number of genotypes exceeds the number of phenotypes. We know that redundant representations increase EA performance if solutions that are similar to the optimal solution are overrepresented (Rothlauf & Goldberg, 2002). The link-and-node biased (LNB) representation proposed by Palmer (1994) is a redundant representation that overrepresents solutions that are similar to the MST (Gaube & Rothlauf, 2001). Consequently, when using this representation for OCST problems, EA performance can be increased. It was shown in Rothlauf (2002) that EAs using the LNB encoding outperform EAs using other representations. The LNB results in higher EA performance in comparison to other problem representations as it is biased towards the MST.

Another possibility to increase EA performance for the OCST problem is to use genetic search operators that favor MST-like trees. For example, Raidl and Julstrom (2003) proposed the edge set representation that is a direct representation of trees for the degree constrained tree problem. Additional heuristics have been introduced for recombination and mutation operators that prefer edges of lower cost. As a result the genetic operators favor solutions that are similar to the MST. Therefore, EAs using such operators show higher performance in comparison to other representation/operator combinations.

A final possibility to increase the performance of EA methods for the OCST problem is to bias the fitness evaluation of trees. If individuals that are similar to the MST get an additional bonus the population converges faster to solutions that are MST-like and the performance of EAs can be increased. Currently, the authors are not aware of any example approaches based on this technique.

It is important to bear in mind that EAs that favor MST-like solutions are only beneficial if it is known that the optimal solution is similar to the MST. This paper has shown that on average OCST problem show this behavior. However, if there is no such problem-specific knowledge, it makes no sense to use the proposed techniques.

4 Future Research

Based on this study some topics require further investigation.

In this work we analyzed the OCST problem and showed statistically that the optimal solution is similar to the MST. An interesting direction of further research is to investigate if the optimal solutions for other constrained tree optimization problems are also similar to the MST. Our belief is that many other constraint tree problems will have similar properties. If this assumption is correct, it would become possible to consider this problem-specific knowledge in heuristic search methods and allow more efficient optimization methods for constrained graph problems to be designed.

In section 3.3 we discussed how the performance of EAs for the OCST problem can be increased by using biased representations, operators, or fitness evaluation techniques. In Rothlauf and Goldberg (2002) we developed a theoretical framework on how biased representations influence EA performance. Using this framework quantitative predictions on the expected EA performance become possible. Until now not much applicable theory exists on how biased operators or fitness evaluation techniques influence EA performance. If models can be developed that give quantitative predictions on EA performance, systematic design of high-quality operators and evaluation functions that consider problem-specific knowledge would become possible. Finding proper operators and evaluation functions would not be a matter of trial-and-error but could become a systematic engineering task.

Peleg and Reshef (1998) presented a deterministic algorithm that constructs a minimum communication spanning tree for 2-dimensional Euclidean trees in $O(\log^2(|V|))$. No problem-specific information about the structure of the optimal solution was used for this construction algorithm. The idea is to develop faster construction algorithms for n-dimensional trees with Euclidean or random distance weights using the knowledge that optimal solutions for the OCST problem are similar to the MST.

5 Summary and Conclusions

This paper started with a short introduction into the optimal communication spanning tree (OCST) problem. Various test instances from the literature are examined and the properties of randomly created solutions are compared to the properties of the best known solutions. Then, a statistical analysis on randomly generated OCST problem instances was performed. Again, the properties of randomly created solutions were compared to the properties of the optimal solutions. The optimal solutions were assumed to be the best solutions that could be found by an evolutionary algorithm (EA) framework. Furthermore, the paper discussed how problem-specific knowledge about the properties of optimal solutions can be considered for the design of representations, search operators, and fitness evaluation methods in EAs. Finally, some directions of further research were presented.

This paper investigates the properties of the OCST problem. It shows that the distance between a randomly created solution and the minimum spanning tree (MST) that is calculated according to the distance weights is on average much higher than the distance between the optimal solution and the MST. This means for the OCST problem that the optimal solution is biased towards the MST. Consequently, the performance of optimization methods for OCST problems can be increased if they are biased towards MST-like solutions.

The bias of the optimal solution towards the MST has been observed for all test instances from the literature that have been investigated. To gain more general results a statistical analysis of OCST problems using either Euclidean or random distance weights has been performed. The results show a strong bias of the optimal solution towards the MST for both, Euclidean and random distance weights.

We strongly encourage users and researchers to consider the presented results for the design of optimization methods for OCST problems. Using optimization methods that show a bias towards MST-like solutions allows more efficient problem solving. Furthermore, we recommend examining whether the optimal solutions for other constrained tree optimization problems are also biased towards the MST. To consider problem-specific knowledge about the problem at hand would allow increase in performance, reliability, and efficiency of the used optimization method.

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.

Berry, L. T. M., Murtagh, B. A., & Sugden, S. J. (1994). A genetic-based approach to tree network synthesis with cost constraints. In Zimmermann, H. J. (Ed.), *Second European Congress on Intelligent Techniques and Soft Computing - EUFIT'94*, Volume 2 (pp. 626–629). Promenade 9, D-52076 Aachen: Verlag der Augustinus Buchhandlung.

Cahn, R. S. (1998). *Wide area network design, concepts and tools for optimization.* San Francisco: Morgan Kaufmann Publishers.

Cayley, A. (1889). A theorem on trees. *Quarterly Journal of Mathematics*, *23*, 376–378.

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*, 236–249.

Davis, L., Orvosh, D., Cox, A., & Qiu, Y. (1993). A genetic algorithm for survivable network design. In Forrest, S. (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp. 408–415). San Mateo, CA: Morgan Kaufmann.

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

Gaube, T., & Rothlauf, F. (2001). The link and node biased encoding revisited: Bias and adjustment of parameters. In Boers, E. J. W., Cagnoni, S., Gottlieb, J., Hart, E., Lanzi, P. L., Raidl, G. R., Smith, R. E., & Tijink, H. (Eds.), *Applications of evolutionary Computing: Proc. EvoWorkshops 2001* (pp. 1–10). Berlin: Springer.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

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., E., G., Wu, A., B., L. W., 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.

Harik, G. R., Cantú-Paz, E., Goldberg, D. E., & Miller, B. L. (1997). The gambler's ruin problem, genetic algorithms, and the sizing of populations. In Bäck, T. (Ed.), *Proceedings of the Forth International Conference on Evolutionary Computation* (pp. 7–12). New York: IEEE Press.

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

Julstrom, B. A. (2001, 7 July). The blob code: A better string coding of spanning trees for evolutionary search. In Wu, A. S. (Ed.), *Proceedings of the 2001 Genetic and Evolutionary Computaton Conference Workshop Program* (pp. 256–261). San Francisco, California, USA.

Kershenbaum, A. (1993). *Telecommunications network design algorithms*. New York: McGraw Hill.

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.

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

Peleg, D., & Reshef, E. (1998). Deterministic polylog approximation for minimum communication spanning trees. *Lecture Notes in Computer Science*, *1443*, 670–682.

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

Raidl, G. R. (2001, February). Various instances of optimal communication spanning tree problems. personal communciation.

Raidl, G. R., & Julstrom, B. A. (2000). A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem. In Carroll, J., Damiani, E., Haddad, H., & Oppenheim, D. (Eds.), *Proceedings of the 2000 ACM Symposium on Applied Computing* (pp. 440–445). ACM Press.

Raidl, G. R., & Julstrom, B. A. (2003). Edge-sets: An effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*. to be published.

Rothlauf, F. (2002). *Representations for genetic and evolutionary algorithms*. Number 104 in Studies on Fuzziness and Soft Computing. Berlin: Springer.

Rothlauf, F., & Goldberg, D. E. (2002). *Redundant representations in evolutionary computation* (IlliGAL Report No. 2002025). Urbana, IL: University of Illinois at Urbana-Champaign. accepted for publication in Evolutionary Computation journal.

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.

Sinclair, M. C. (1995). Minimum cost topology optimisation of the COST 239 European optical network. In Pearson, D. W., Steele, N. C., & Albrecht, R. F. (Eds.), *Proceedings of the 1995 International Conference on Artificial Neural Nets and Genetic Algorithms* (pp. 26–29). New York: Springer-Verlag.

Tang, K. S., Man, K. F., & Ko, K. T. (1997). Wireless LAN desing using hierarchical genetic algorithm. In Bäck, T. (Ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms* (pp. 629–635). San Francisco: Morgan Kaufmann.