Guided Local Search for the Optimal Communication Spanning Tree Problem

Wolfgang Steitz, Franz Rothlauf
Dept. of Information Systems and Business Administration
University of Mainz, Germany
steitzw|rothlauf@uni-mainz.de

ABSTRACT

This paper considers the optimal communication spanning tree (OCST) problem. Previous work analyzed features of high-quality solutions. Consequently, integrating this knowledge into a metaheuristic increases its performance for the OCST problem. In this paper, we present a guided local search (GLS) approach which dynamically changes the objective function to guide the search process into promising areas. In contrast to traditional approaches which reward promising solution features by favoring edges with low weights pointing towards the tree's center, GLS penalizes low-quality edges with large weights that do not point towards the tree's center.

Categories and Subject Descriptors: I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search

General Terms: Algorithms, Design

Keywords: optimal communications spanning tree, guided local search, problem-specific adaptation

1. INTRODUCTION

The optimal communication spanning tree (OCST) problem is a common combinatorial tree optimization problem [3]. Given a collection of nodes and the distances and communications demands between them, we seek a tree that connects all nodes at minimum total communications cost, defined to be the sum over all pairs of nodes of the products of the distances and communications demands between them.

The OCST problem is $\mathcal{NP}$-hard [2] and $\mathcal{MAX}$ $\mathcal{SNP}$-hard. Researchers studied various exact, approximate and heuristic solution approaches for the problem. The current state-of-the-art approaches [4, 1, 5, 6] are based on heuristics and metaheuristics, in particular evolutionary algorithms.

Previous work [6, 5] studied properties of OCST problems. They found that edges with low distance weights pointing towards the center of a graph occur in high-quality solutions with higher probability. There are several possibilities of how such a-priori knowledge can be exploited in the design of efficient metaheuristics for OCST problems. In this paper, we apply GLS to the OCST problem and show how problem-specific knowledge can be used to improve the search performance by changing the evaluation function.

Copyright is held by the author/owner(s).
GECCO'11, July 12–16, 2011, Dublin, Ireland.
ACM 978-1-4503-0690-4/11/07.

2. GUIDED LOCAL SEARCH

Guided local search (GLS) [8], ensures diversification by dynamically changing the objective function by adding penalties. This allows search to escape local optima. GLS's mechanism to change the objective function is based on *solution features*. Since edges are proper solution features for graph based problems [8], we define the set of solution features for the OCST problem as all edges in a fully connected graph.

At the beginning of a GLS run, a starting solution T_0 is created and all penalties are set to zero. Then, a local search method using the original evaluation function f is started from solution T_0 and a local optimum T_1 is found. Now the penalties for solution features which cause the highest costs are incremented by 1. The higher the cost of a solution feature, the higher the utility of penalizing this feature. For the OCST problem, the cost of a solution feature is simply the weight of an edge or the orientation of an edge or calculated using weight and orientation. To ensure diversification and to avoid always the same features being penalized, the current penalties are part of the utility function, which calculates the utility of penalization.

After the initialization phase, the optimization phase is iterated until a previously defined stopping criterion is reached. First, a local search method is started from solution T_k and a new local optimum T_{k+1} is found. Then, some solution features of the current local optima are penalized. The local search in the optimization phase uses the altered objective function $h(T) = f(T) + \lambda \cdot \sum_{i=1}^{M} p_i \cdot I_i(T)$, where $f(T)$ is the original evaluation function with no modifications, M is the number of solution features, λ is a parameter that controls the importance of the penalties. Since we use the edges as solution features, $M = n - 1$ for the OCST problem. Using h, the current solution is not attractive anymore, the local search method is able to escape from the local optimum.

Search performance strongly depends on the setting of λ. Tsang et. al. [7] suggested to calculate the parameter according to the first found local optima T_1 as $\lambda = \alpha \cdot f(T_1)/M$. The only parameter required tuning is $\alpha \in]0, 1]$. In our experiments best results are achieved with a setting of $\alpha = 0.3$.

3. EXPERIMENTAL RESULTS

Following previous work [4, 5] we use randomly created OCST test instances. The real-valued demands r_{ij} are randomly created and uniformly distributed in $]0, 10]$. The distance weights w_{ij} are calculated as the Euclidean distances between the nodes i and j which are randomly placed on a 10×10 2-dimensional grid.

We compare the GLS approach to an EA using edge-sets

Table 1: Performance comparison

n	$eval$		ES-w	ES-o	ES-ow	GLS-w	GLS-o	GLS-ow
10	2k	P_{suc}	0.43	0.14	0.67	**0.74**	0.7	0.71
		μ	1498.55	1535.0	**1492.37**	1494.96	1496.2	1495.49
		σ	7.2	20.28	3.42	11.34	12.9	11.81
15	3k	μ	3675.87	3881.14	**3617.82**	3622.12	3622.94	3624.59
		σ	32.06	77.71	15.77	45.04	45.25	46.9
25	5k	μ	10994	12336	10616	**10599**	10599	10599
		σ	149.33	439.71	76.89	198.46	188.21	188.34
50	10k	μ	48694	60946	45900	45578	45581	**45539**
		σ	996.33	3273.73	647.76	1244.97	1226.11	1210.7
75	20k	μ	112189	136220	105026	103598	**103432**	103485
		σ	2298.56	7053.38	1598.57	2657.63	2600.17	2514.29
100	40k	μ	199753	227468	187148	**185208**	185903	185479
		σ	3914.58	8416.62	2404.58	4311.28	4886.46	4654.08

[4] (ES). The population size $N = 200$. We extend the original crossover operator, which only considers edge weights, and consider edge weights *and* edge orientations as proposed in [6]. Mutation is applied after crossover with probability $p_m = 1/n$. The initial population consists of random spanning trees.

For both GLS and ES we study three different variants of GLS using different costs of the solutions features: 1) only edge weights (-w), 2) only edge orientation (-o) , and 3) edge weight and edge orientation (-ow). Each optimization run is terminated after $eval$ fitness evaluations. Since a larger number of evaluations increases search performance, we also increase $eval$ with larger n (see Table 1). For each OCST problem instance, ten optimization runs are performed.

Table 1 shows P_{suc} (only for $n = 10$), the mean cost μ of the best solution at the end of a run and the corresponding standard deviation σ. We observe that all GLS variants outperform ES for larger problem instances ($n \geq 25$). For smaller instances, ES is slightly better. Comparing the different ES configurations, we find that considering problem-specific knowledge for the crossover operator improves search performance. The more knowledge is integrated, the better are the results. ES-ow is significantly better than ES-o or ES-w. The situation is different for GLS. The average fitness μ at the end of a run shows no significant difference whether we use edge weights, edge orientation or both as costs of solution features.

The main difference between ES and GLS is the type of additional selection pressure that is introduced by considering problem-specific knowledge. ES favors edges with low weight and proper orientation by giving them a higher chance to be selected in recombination. In contrast, GLS penalizes low-quality solution features and does not reward "good" solution features. Since all three types of feature costs (weight, orientation, and both combined) are able to identify low-quality solutions features, we observe no performance difference. We study this conjecture in the further paragraphs.

Figure 1 plots the average weight w_{ij} (left) and the average edge orientation γ_{ij} (right) of the edges in the current best found solution over the number of search steps.

For ES, the choice of problem-specific knowledge has a strong influence on the properties of the found solution. When considering only orientation (ES-o), an ES run finds good solutions with many edges that point towards the graph center (low γ). Edge weights are also reduced, but less in comparison to other variants or in comparison to the change

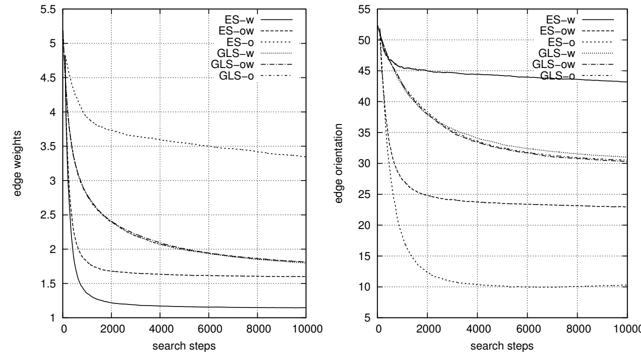

Figure 1: Average edge weight (left) and orientation (right) over the number of search steps for $n = 50$.

of γ. When considering only edge weights (ES-w), we observe the opposite behavior: ES finds solutions with low-weight edges , which do not necessarily point towards the center of the graph. If we consider both weights and orientation (ES-ow), we are able to find solutions with low-weight edges that point towards the graph center, however less than when considering only edge weights (ES-w) or orientation (ES-o), respectively. In contrast, for GLS the choice of the cost of solution feature has no significant influence. All three variants show similar behavior.

4. OBSERVATIONS

Problem-specific GLS outperforms state-of-the-art EAs using edge-sets although EAs also consider weights and orientation of edges. The performance of GLS is about independent of how much problem-specific knowledge is considered. Thus, GLS shows similar behavior if either edges with high weight, wrong orientation, or both are penalized. This is in contrast to results for EAs with edge-sets where the choice of problem-specific knowledge has a strong influence on search performance.

References

[1] T. Fischer and P. Merz. A memetic algorithm for the optimum communication spanning tree problem. *Lecture Notes in Computer Science*, 4771:170, 2007.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.

[3] T. C. Hu. Optimum Communication Spanning Trees. *SIAM Journal on Computing*, 3(3):188–195, September 1974.

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

[5] F. Rothlauf. On Optimal Solutions for the Optimal Communication Spanning Tree Problem. *Operations Research*, 57(2):413–425, 2009.

[6] W. Steitz and F. Rothlauf. Edge orientation and the design of problem-specific crossover operators for the ocst problem. *IEEE Transactions on Evolutionary Computation*, forthcoming, 2011.

[7] E. Tsang, P. Mills, and J. Ford. Extending guided local search – towards a metaheuristic algorithm with no parameters to tune. Technical Report 371, Dep. of Comp. Sci., University Essex, 2002.

[8] C. Voudouris. Guided local sarch. In F. Glover and G. A. Kochenberger, editors, *Handbook of Metaheuristics*, pages 185–218. Springer, 2003.