2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

A Neighborhood Structure for the CMST Problem

Jun Han+, Ming Lei, Xiang Long, Dianfu Ma, Xin Wei

School of Computer Science and Engineering, Beihang University, China

{jun_han, long, dfma}@buaa.edu.cn, {leiming, weixin} @act.buaa.edu.cn

Abstract. This paper studies the capacitated minimum spanning tree (CMST) problem, which is one of the most fundamental and significant problems in the optimal design of communication networks. A new neighborhood structure is proposed by analyzing the features of the optimal solution of the CMST problem. A new neighborhood search algorithm based on the new neighborhood structure and certain new searching methods is designed and implemented. Computational experiments showing the effectiveness of the new algorithm on benchmark instances are given.

Keywords: CMST, Heuristic Algorithm, Neighborhood Search, Tabu Search

1. Introduction

Given a centralized undirected graph with costs associated with its edges, the capacitated minimum spanning tree problem is to find a minimum cost spanning tree of the given graph, subject to a capacity constraint in all subtrees incident to the central node. It is one of the most fundamental problems in the optimal design of communication networks, as well as general network optimizations. It is widely used as a model in a star network system to make it adaptive and economical. Moreover, it is also one classical type of general model or optimization, globally or locally. The cost established on arcs may represent kinds of meanings, such as the communication cost or interdependence coefficients among network nodes, geographic nodes, logistics nodes, etc. The CMST problem is NP-hard when $2 < K < n/2$, as proved by Papadimitriou in [15].

The existing exact algorithm only solve some small scale problems [9][10][13]. The heuristic algorithm will be the best choice when solve the large scale CMST problem. There are kinds of heuristic algorithms, such as taboo search [3][17], ant colony algorithm [16], genetic algorithms [4][18], neighborhood search [1][2][3][11][17] and second order algorithm [14] etc.

Generally, a neighborhood search algorithm starts with a feasible solution S. Following some predefined neighborhood structure, the algorithm generates neighbors of S, by performing certain changes, and then updates S with one of its neighbors according to certain routines. The process is repeated until a termination condition is satisfied.

2. Neighborhood Structures

2.1. Existing Neighborhood Structures

The performance of a neighborhood search algorithm crucially depends on the neighborhood structure.

In Amberg et al.'s work [3], the neighborhood search method bases a neighborhood "move" on transforming an existing feasible solution to another. The neighborhood structure is considered as a two-exchange structure, which defines that at most two substrees are exchanged for each "move". Two types of moves are defined as Node Shift and Node Exchange, the former move chooses one node and moves it to a

+ Corresponding author. Tel.: +8601082316340; fax: +8601082316340.
E-mail address: jun_han@buaa.edu.cn.

subtree which differs from the one it belongs to, while the latter move chooses two nodes from two different subtrees and exchanges them.

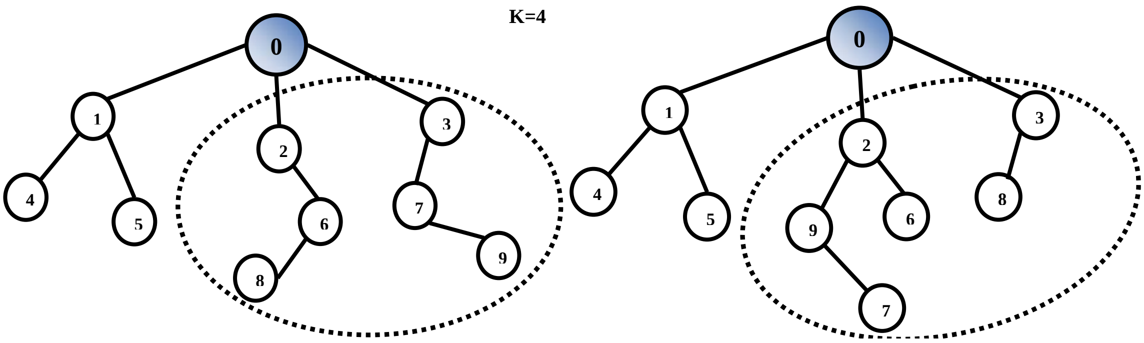
In Sharaiha et al.'s work [7], the neighborhood search method bases on "cut and paste" operations. The "cut" operation may cut either a whole subtree or a part, and the "paste" operation connects the cutted subtree to the center or another subtree. Ahuja has pointed that this "cut and paste" operation may also be viewed as a two-exchange neighborhood structure, see Figure 2. The complexity of the method for determine the most profitable cut and paste operation is O(N2K2).

Ahuja et al. proposed a composite neighborhood structure [1], which includes both the node-based and tree-based neighborhoods he proposed in [2]. In the neighborhood search, cyclic exchanges are performed, which involves several rooted subtrees where each rooted subtree contributes a single node or a small subtree. The cyclic exchange is defined as a valid cyclic if it decreases the total cost. In worst case, the time complexity of the cyclic exchange is O((N/K)K), which is too time expensive. To simplify the search, they propose a heuristic with time complexity O(|N||E|K2K) to find the certain cycles. The Computational results reported show the dominant of the neighborhood search algorithm for CMST on standard benchmark instances.

Han et al.'s neighborhood structure [11] is similar to Sharaiha et al's. They both find the solution's neighbors by moving sub-trees. Being different with Sharaiha et al's algorithm, Han et al.'s subtree moving often involves more than two rooted subtrees. When a subtree move results in the violation of the capacity constraints within a rooted subtree, a new subtree will be moved out from the underlying root subtree and will be moved into another rooted subtree. This sub-procedure continues until every rooted subtree satisfies the capacity constraint, and the process is called a "move". In order to make sure that no endless moving cycles occur, the algorithm restraints that a rooted subtree can only be changed at most once in a "move". Obviously, this algorithm's neighborhood space is larger than Sharaiha et al's, so it could find better solutions.

2.2. Component Pair Optimization Neighborhood Structure

In this paper, we propose a "Component Pair Optimization" neighborhood structure. We define a minimum cost spanning tree over part of the nodes, with the same capacity constraints and the same central node is a sub-CMST problem of the original CMST problem. It can be proved that, if a solution tree T is the optimal solution of a CMST problem, then every pair of rooted subtrees (components) is the optimal solution of the sub-CMST problem which covers all the nodes of the two root subtrees. We propose a new neighborhood structure, namely component pair optimization. Whenever a pair of components is replaced by the optimal solution tree of the sub-CMST problem which covers all the nodes of this pair of components, a neighbor of the current tree solution of the original problem is generated. If the capacity of a CMST problem is K, the size of its sub-CMST problem is no more than 2K. Since K is usually small, it is easy to solve the sub-CMST problem to optimality. Hence this neighborhood structure is applicable for the CMST problem whose capacity constraint is small, which is the case of generally believed hard problems. Figure 1 shows the proposed neighborhood structure.



(a). The tree solution before component pair (b). The tree solution after component pair

Fig. 1: Component Pair Optimization Neighborhood Structure

3. The Neighborhood Search Algorithm

3.1. Initial Solution Algorithm

We used a randomized size-first version of the Esau-William's algorithm, which is perhaps the most popular construction based heuristic algorithm for the capacitated minimum spanning tree problem.

The Esau-William algorithm [5] starts with each subtree containing a singleton node. In each iteration, the algorithm joins two subtrees into a single subtree so that the new subtree satisfies the capacity constraints and the savings achieved by the join operation are maximized.

In this randomized size-first version of Esau-William algorithm, we determine the p biggest size profitable join operations for some small value p in each iteration. We then generate an integer random number k uniformly distributed between 1 and p and perform the kth biggest size profitable join operation. This method in general provides a new feasible tree each time it is applied. Since at each step it performs one of the p biggest size profitable join operations, the feasible tree obtained is generally a good tree. In our algorithm, we used p .

3.2. Solving the Sub-CMST Problem

In order to find the neighbors of the current solution, some sub-CMST problems have to be solved to optimality. Although the CMST problem is NP-hard, it is still practical to find the optimal solution of a CMST problem by an exact algorithm when the size of the problem is small enough. In our new neighborhood search algorithm, we employ the node-orient branch and bound algorithm proposed by Han et al. [10], which is known to be the fastest exact algorithm for solving CMST, to acquire the optimal solution of the sub-CMST problem. Let K be the capacity constraint of the targeting CMST problem, then the problem size of our sub-CMST problem is no more than 2K. When $K \leq 10$, we get $2K \leq 20$, which is the size of problem that can be optimized by Han et al.'s exact algorithm very quickly.

3.3. The Main Search Procedure of the Neighborhood Search Algorithm

The main procedure of our neighborhood search algorithm is given below. The algorithm starts with a feasible solution T, which is referred to as the "current solution" at the beginning, obtained by randomized size-first Esau-William algorithm. It checks if there is a neighbor T' with least cost by solving sub-CMST problem. If yes, T' becomes the current solution. This process is continued until no better neighbor can be found. When the above process terminates the algorithm completes one run. In the next run, the algorithm starts with another feasible solution obtained again by the randomized size-first Esau-William algorithm and applies the search process again. The algorithm terminates when the total time taken by the algorithm has reached a specified upper bound. The main procedure is as shown in Figure 2.

```
while (time<set_time) {
    bestcost = MAX;
    bestsolution = NULL;
    T = initialsolution();
    while (T has component pair that could be optimized ) {
        st=get_component_pair(T);
        opt_st=optimization(St);
        T=neighbor(T, opt_st);
            }
            if (cost(T) < bestcost){
              bestcost = cost(T);
              bestsolution = T;
            }
      }
```

Fig. 2 Main Search Procedure

4. Computational Results

We applied our algorithm to the benchmark problems with the capacity constraint no more than 10 provided by OR-Library (http://people.brunel.ac.uk/~mastjjb/jeb/info.html). Our program is set to run 10 seconds for tc problems with 41 nodes and 50 seconds for te problems with 41 nodes and 1800 seconds for problems with 81 nodes.

Table 1 is comparisons of our neighborhood search algorithm and the other two neighborhood search algorithms for benchmark problems with 41 nodes. The VLNS algorithm is currently the best heuristic algorithm which was proposed by Ahuja [1]. The integrated neighborhood search algorithm was proposed by Han et al. [11]. It can be seen from Table 1 that, the integrated neighborhood search algorithm is faster when solving the "tc" class problems whose capacity constraint is 10. For the other problems however, our proposed algorithm can give the same results in the least time frame.

Table 2 is the compute results for benchmark problems with 81 nodes. It can be seen that, the new neighborhood search algorithm could find most of the best available solutions. But compared to Ahuja's VLNS, our method could not find the same good solutions for some benchmark problems. The best results are presented in bold face. The starred values are known to be optimal solutions of the corresponding benchmark solutions.

5. Conclusion

In this paper we proposed a novel neighborhood structure for solving the CMST problem. The structure defined a neighbor of a solution tree to be one with its certain component pairs locally optimized. Then a new neighborhood search algorithm based on the new neighborhood structure for the CMST problem is designed and implemented. Computational experiments showed that the proposed algorithm is effective and efficient for solving CMST problems. Future work will be on the further improvement of the neighborhood structure and the search regulation to get the better results of the large scale CMST problems in shorter time.

6. Acknowledgments

This work was funded by the National Natural Science Foundation of China (NSFC grant number: 60842008 and 90818028). The authors would also like to thank all those who took part

in the focus groups and experiments.

Table 1: Comparisons with other neighborhood search algorithms.

Problem Name	Capacity	Best available solution	Han,J. et al.		VLNS		Proposed algorithm	
			Solution	Time(s)	Solution	Time(s)	Solution	Time(s)
tc41-1	3	742*			**742**	100	**742**	10
tc41-2	3	717*			**717**		**717**	
tc41-3	3	716*			**716**		**716**	
tc41-4	3	775*			**775**		**775**	
tc41-5	3	741*			**741**		**741**	
tc41-1	5	586*			**586**		**586**	
tc41-2	5	578*			**578**		**578**	
tc41-3	5	577*			**577**		**577**	
tc41-4	5	617*			**617**		**617**	
tc41-5	5	600*			**600**		**600**	
tc41-1	10	498*	**498**	1	**498**		**498**	
tc41-2	10	490*	**490**		**490**		**490**	
tc41-3	10	500*	**500**		**500**		**500**	
tc41-4	10	512*	**512**		**512**		**512**	
tc41-5	10	504*	**504**		**504**		**504**	
te41-1	3	1190*			**1190**		**1190**	50
te41-2	3	1103			**1103**		**1103**	
te41-3	3	1115*			**1115**		**1115**	
te41-4	3	1132*			**1132**		**1132**	
te41-5	3	1104*			**1104**		**1104**	

te41-1	5	830*			**830**		**830**	
te41-2	5	792*			**792**		**792**	
te41-3	5	797*			**797**		**797**	
te41-4	5	814*			**814**		**814**	
te41-5	5	784*			**784**		**784**	
te41-1	10	596*	**596**	1000	**596**		**596**	
te41-2	10	573*	**573**		**573**		**573**	
te41-3	10	568*	**568**		**568**		**568**	
te41-4	10	596*	598		**596**		**596**	
te41-5	10	572*	**572**		**572**		**572**	

Table 2: Computational results for 81 nodes problem

Problem	Capacity	Best available	VLNS	Proposed
tc81-1	5	1099*	**1099**	**1099**
tc81-2	5	1100*	**1100**	**1100**
tc81-3	5	1073*	**1073**	**1073**
tc81-4	5	1080*	**1086**	**1080**
tc81-5	5	1287*	**1287**	**1287**
tc81-1	10	888*	**888**	**888**
tc81-2	10	877*	**877**	**877**
tc81-3	10	878*	**878**	**878**
tc81-4	10	868*	**868**	**868**
tc81-5	10	1002*	**1002**	**1002**
te81-1	5	2544*	**2544**	**2544**
te81-2	5	2551*	**2551**	2554
te81-3	5	2612*	**2612**	**2612**
te81-4	5	2558*	**2558**	**2558**
te81-5	5	2469*	**2469**	**2469**
te81-1	10	1657*	**1657**	**1657**
te81-2	10	1639*	**1639**	1645
te81-3	10	1687*	**1687**	**1687**
te81-4	10	1629*	**1629**	**1629**
te81-5	10	1603*	**1603**	**1603**

7. References

[1] Ahuja, R.K., J. B. Orlin and D. Sharma. A composite very large-scale neighborhood structure for the capacitated minimum spanning tree problem [J]. *Operations Research Letters*, 2003, v31:185-194.

[2] Ahuja, R.K., Orlin, J.B. and Sharma, D. Multi exchange neighborhood search algorithms for the capacitated minimum spanning tree problem [J]. *Mathematical Programming*, 2001, V91: 9-39.

[3] Amberg A., W. Domschke, and S. Vob. Capacitated minimum spanning trees: intelligent search[J]. *Combinatorial Optimization: Theory and Practice,* 1996, V1: 9-39.

[4] De Lacerda, Estefane George Macedo. A genetic algorithm for the capacitated minimum spanning tree problem [A]. *IEEE Congress on Evolutionary Computation* [C], 2006:725-729.

[5] Esau, L.R. and Williams, K.C. On Teleprocessing System Design [J]. *IBM Systems Journal*, 1966, V5:142-147.

[6] Gavish, B. Topological design of telecommunication networks-local access design methods[J]. *Annals of Operations Research*, 1991, V33: 17-71.

[7] Gavish, B. and Altinkemer, K. Parallel savings heuristics for the topological design of local access tree networks [A]. Proc. *IEEE INFOCOM 86 Conference*[C], 1986: 130-139.

[8] Gavish, B. Formulations and algorithms for the capacitated minimal directed tree problem[J]. *Journal of the ACM*, 1983, V30: 118-132.

[9] Han, J., McMahon, G.B. and Sugden, S.J. A branch and bound algorithm for capacitated minimum spanning tree problem [J]. *Lecture Notes in computer Science*, 2002, V2400: 404-408.

[10] Han, J., McMahon, G.B. and Sugden, S.J. A Node-oriented Branch and Bound Algorithm for the Capacitated

Minimum Spanning Tree Problem [A]. *IEEE Computer Society*[C], 2003: 460-469.

[11] Han, J., Sugden, S.J. and Jinpeng, Huai. A Neighborhood Search Algorithm for the Capacitated Minimum Spanning Tree Problem [J]. *Working paper, BUAA.* http://www.act.buaa.edu.cn/tutor_HanJun_2.asp, 2008

[12] Kershenbaum, A., Boorstyn, R.R. and Oppenheim, R. Second-order greedy algorithms for centralized teleprocessing network design [A]. *IEEE Transactions on Communications*[C], 1980, V28: 1835 - 1838.

[13] Malik, K. and Yu, G. A branch and bound algorithm for the capacitated minimum spanning tree problem [J]. *Networks,* 1993, V23: 525-532.

[14] Martins, P. Enhanced second order algorithm applied to the capacitated minimum spanning tree problem [J]. *COMPUT OPER RES,* 2007, V34 (8): 2495-2519

[15] Papadimitriou, C.H. The Complexity of the capacitated tree problem [J]. *Networks,* 1978, V8: 217-230.

[16] Reimann, Marc. Savings based ant colony optimization for the capacitated minimum spanning tree problem [J]. *Computers and Operations Research,* 2006, V33(6):1794-1822.

[17] Sharaiha, Y. M., M. Gendreau, G. Laporte, and I. H. Osman. A tabu search algorithm for the capacitated shortest spanning tree problem [J]. *Networks,* 1997, V29:161-171.

[18] Thangiah, S.R., Osman, I.H. and Sun, T. Hybrid genetic algorithms, simulated annealing and tabu search methods for vehicle routing problems with time windows. *Working Paper,* 1994, Univ. of Kent, Canterbury.