Enumeration of Pareto Optimal Multi-Criteria Spanning Trees - A Proof of the Incorrectness of Zhou and Gen's Proposed Algorithm

Joshua D. Knowles *

David W. Corne†

Department of Computer Science, University of Reading, UK.

*Contact details: IRIDIA CP 194/6, ULB, Avenue Franklin D. Roosevelt 50, 1050 Brussels, Belgium. Email: jknowles@ulb.ac.be, Tel: +32 (0)2 650 31 70, Fax: +32 (0)2 650 27 15

†Email: d.w.corne@reading.ac.uk

Keywords: Combinatorial optimization; minimum spanning tree; graph theory; multiple-criteria optimization; genetic algorithms.

Abstract

The minimum spanning tree (MST) problem is a well-known optimization problem of major significance in operational research. In the *multi-criteria* MST (mc-MST) problem, the scalar edge weights of the MST problem are replaced by vectors, and the aim is to find the complete set of Pareto optimal minimum-weight spanning trees. This problem is NP-hard and so approximate methods must be used if one is to tackle it efficiently. In an article previously published in this journal, a genetic algorithm (GA) was put forward for the mc-MST. To evaluate the GA, the solution sets generated by it were compared with solution sets from a proposed (exponential time) algorithm for enumerating all Pareto optimal spanning trees. However, the proposed enumeration algorithm that was used is not correct for two reasons: 1. It does not guarantee that all Pareto optimal minimum-weight spanning trees are returned; 2. It does not guarantee that those trees that are returned are Pareto optimal. In this short paper we prove these two theorems.

1 Statement of the problem

The multi-criteria minimum spanning tree (mc-MST) problem can be simply stated. Given a weighted graph $G = (V, E)$ with vertex set V and edge set E and edge weight vectors $w_i(e) \in \Re^+, e \in E, i \in 1..K$ where K is the number of criteria, find a spanning tree T in G such that there does not exist another spanning tree whose total weights Pareto dominate T.

2 Zhou and Gen's Proposed Enumeration Algorithm

In an article by Zhou and Gen [1], an algorithm for enumerating all Pareto optimal spanning trees was proposed. The operation of the algorithm can be summarised as follows:

Step 1: Pick an arbitrary start-vertex v_1. In turn, consider each edge adjacent to v_1. Put all edges that are nondominated into a set of subtrees S.

Step 2: For each subtree $s \in S$ consider, in turn, each adjacent edge that does not cause a cycle to be created when added to s. For each such edge that can be added to s to form a new subtree t which is nondominated by any other subtree sprouting from s, put t into a new set T.

Step 3: $S \leftarrow T$, $T \leftarrow \emptyset$. Compact down the set of subtrees S such that all dominated subtrees, and all repeated subtrees, are removed.

Step 4: If the subtrees in S have $V - 1$ edges then S is the required set of unique, Pareto optimal spanning trees. Else return to Step 2.

In [1] the above algorithm was used to enumerate the set of Pareto optimal spanning trees on some small mc-MST instances. The resulting solution sets were then used to measure the proportion of solutions that were Pareto optimal from those generated by the authors' proposed genetic algorithm. In the next section we prove that the enumerative algorithm is incorrect: it neither guarantees returning all Pareto optimal solutions, nor that those returned are Pareto optimal.

3 Incorrectness of the Proposed Approach

Theorem: *The proposed algorithm is not guaranteed to generate* all *Pareto optimal spanning trees.*

Proof: The proof is by example. Consider a 4-vertex weighted complete graph G_1 in which each edge has two weights associated with it as shown in Figure 1.

[Figure 1 goes here]

In total there are 16 spanning trees of the graph G_1, of which 6 are Pareto optimal. The complete list of spanning trees of G_1 is given in Table 1 with the 6 Pareto optimal spanning trees shown diagrammatically.

[Table 1 goes here]

Given G_1 and a start-vertex of 0, the algorithm of Zhou and Gen generates the trace given in Table 2. The trace is interpreted as follows: The left hand column lists all the edges that can be added to $v_1 = 0$, and next to each listed edge, its corresponding weight vector is given. Since all of these edges are nondominated each becomes a subtree in S, to which other edges can be joined. The second column shows each of the edges that can be added to each of the subtrees in column 1, each forming a new subtree of two edges. Similarly, for each subsequent column, the edges that can be added to the subtree of the previous column are listed. Next to each subtree in the trace the *total* vector subtree weight is given. A double-asterisked subtree denotes one which is dominated by other subtrees sprouting from the same subtree at the previous level, and a single-asterisked solution denotes one which is dominated by a subtree sprouting from a different sub-tree at the previous level. Both single and double-asterisked solutions are discarded in Zhou and Gen's algorithm, and no further edges are added to them. The final solutions returned by the algorithm are thus those in the right column with no asterisks. A repeated tree, i.e. one with identical edges to another one, is denoted by an "R". The trace shows that the algorithm generates only 4 of the 6 Pareto optimal spanning trees of the graph G_1. □

[Table 2 goes here]

Theorem: *The algorithm of Zhou and Gen can generate spanning trees that are not Pareto optimal.*

Proof: The proof is by example. Consider a 5-vertex weighted complete graph G_2 in which each edge has two weights associated with it as shown in Figure 2.

[Figure 2 goes here]

Then given G_2 and a start-vertex of 0, the algorithm of Zhou and Gen generates the trace given in Table 3. The generated spanning tree 0-3,1-4,2-3,3-4 with total weight vector = (1987,1054) is not dominated by any other generated in the trace, and is thus returned as an optimal solution. However, this spanning tree is dominated by the feasible spanning tree 0-2,1-2,2-3,3-4 (shown in Figure 3) which has a weight vector = (1898,1021). □

[Table 3 goes here]

[Figure 3 goes here]

4 Conclusion

Since the algorithm of Zhou and Gen fails both to guarantee returning all Pareto optimal spanning trees, and that all those generated are Pareto optimal, it cannot be said to be an enumerative procedure for finding Pareto optimal solutions to the mc-MST problem. Therefore, it cannot be validly used as a method for measuring the number (or proportion) of spanning trees which are Pareto optimal from any given set. Thus it should not be used, in future, for evaluating the quality of mc-MST solutions generated by an approximate method such as a genetic algorithm.

References

[1] Gengui Zhou and Mitsuo Gen. Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research*, 114(1), April 1999.

edge	weights
0-1	(3,4)
0-2	(2,6)
0-3	(4,3)
1-2	(1,10)
1-3	(6,2)
2-3	(2,2)

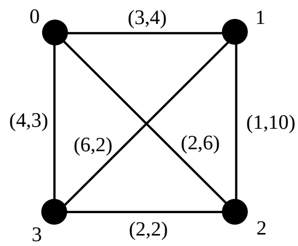


Figure 1: The graph G_1

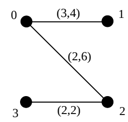
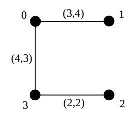
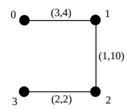
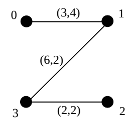
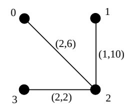
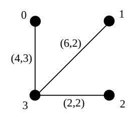
Edges	Total tree weights	Pareto optimal	Figure
0-1,0-2,0-3	(9,13)	no	
0-1,0-2,1-3	(11,12)	no	
0-1,0-2,2-3	**(7,12)**	yes	
0-1,0-3,1-2	(8,17)	no	
0-1,0-3,2-3	**(9,9)**	yes	
0-1,1-2,1-3	(10,16)	no	
0-1,1-2,2-3	**(6,16)**	yes	
0-1,1-3,2-3	**(11,8)**	yes	
0-2,0-3,1-2	(7,19)	no	
0-2,0-3,1-3	(12,11)	no	
0-2,1-2,1-3	(9,18)	no	
0-2,1-2,2-3	**(5,18)**	yes	
0-2,1-3,2-3	(10,10)	no	
0-3,1-2,1-3	(11,15)	no	
0-3,1-2,2-3	(7,15)	no	
0-3,1-3,2-3	**(12,7)**	yes	

Table 1: The spanning trees of G_1 with the Pareto optimal ones indicated in bold and illustrated by a figure

0-1 (3,4)		
	0-2 (5,10)*	
	0-3 (7,7)*	
	1-2 (4,14)*	
	1-3 (9,6)*	
0-2 (2,6)		
	0-1 (5,10)**	
	0-3 (6,9)**	
	1-2 (3,16)	
		0-3 (7,19)**
		1-3 (9,18)**
		2-3 (5,18)
	2-3 (4,8)	
		0-1 (7,12)
		1-2 (5,18)R
		1-3 (10,10)*
0-3 (4,3)		
	0-1 (7,7)**	
	0-2 (6,9)**	
	1-3 (10,5)**	
	2-3 (6,5)	
		0-1 (9,9)
		1-2 (7,15)*
		1-3 (12,7)

Table 2: The trace of the algorithm of Zhou and Gen as it generates spanning trees of the graph G_1, using start-vertex $v_1 = 0$

edge	weights
0-1	(356,979)
0-2	(587,285)
0-3	(745,225)
0-4	(26,603)
1-2	(926,138)
1-3	(826,546)
1-4	(857,231)
2-3	(375,323)
2-4	(422,711)
3-4	(10,275)

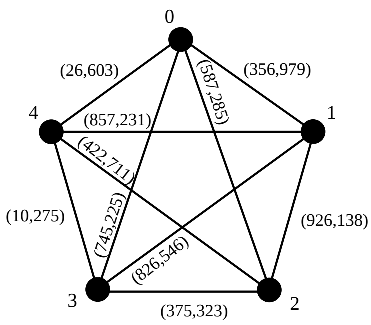


Figure 2: The edge-list of the weighted graph G_2

0-1 (356,979) **			
0-2 (587,285)			
	0-1 (943,1264) **		
	0-3 (1332,510) *		
	0-4 (613,888) *		
	1-2 (1513,423)		
		0-3 (2258,648)	
			0-4 (2284,1251) **
			1-4 (3115,879) *
			2-4 (2680,1359) **
			3-4 (2268,923)
		0-4 (1539,1026) *	
		1-3 (2339,969) **	
		1-4 (2370,654) **	
		2-3 (1888,746) *	
		2-4 (1935,1134) **	
	2-3 (962,608) *		
	2-4 (1009,996) **		
0-3 (745,225)			
	0-1 (1101,1204) **		
	0-2 (1332,510) **		
	0-4 (771,828) **		
	1-3 (1571,771) **		
	2-3 (1120,548) **		
	3-4 (755,500)		
		0-1 (1111,1479) *	
		0-2 (1342,785)	
			0-1 (1698,1764) *
			1-2 (2268,923) *
			1-3 (2168,1331) *
			1-4 (2199,1016) *
		1-3 (1581,1046) **	
		1-4 (1612,731)	
			0-2 (2199,1016) *
			1-2 (2538,869)
			2-3 (1987,1054) R
			2-4 (2034,1442)**
		2-3 (1130,823)	
			0-1 (1486,1802)*
			1-2 (2056,961)
			1-3 (1956,1369)*
			1-4 (1987,1054)
		2-4 (1177,1211) **	
0-4 (26,603)			
	0-1 (382,1582) **		
	0-2 (613,888) **		
	0-3 (771,828) *		
	1-4 (883,834) **		
	2-4 (448,1314) **		
	3-4 (36,878)		
		0-1 (392,1857)	
			0-2 (979,2142)
			1-2 (1318,1995) *
			2-3 (767,2180)
			2-4 (814,2568) **
		0-2 (623,1163)	
			0-1 (979,2142) R
			1-2 (1549,1301)
			1-3 (1449,1709) *
			1-4 (1480,1394) *
		1-3 (862,1424) **	
		1-4 (893,1109)	
			0-2 (1480,1394) *
			1-2 (1819,1247)
			2-3 (1268,1432)
			2-4 (1315,1820) **
		2-3 (411,1201)	
			0-1 (767,2180) R
			1-2 (1337,1339)
			1-3 (1237,1747)
			1-4 (1268,1432) R
		2-4 (458,1589) **	

Table 3: The trace of the algorithm of Zhou and Gen as it generates spanning trees of the graph G_2, using start-vertex $v_1 = 0$

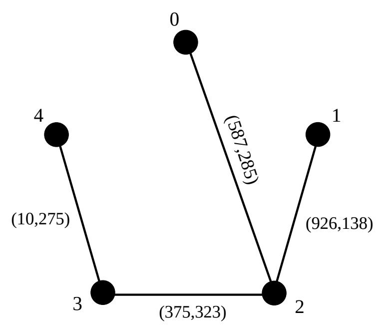


Figure 3: The spanning tree 0-2,1-2,2-3,3-4 in G_2