PERGAMON

Computers & Operations Research 29 (2002) 441–458

computers & operations research

www.elsevier.com/locate/dsw

The cable trench problem: combining the shortest path and minimum spanning tree problems

Francis J. Vasko[a,b,*], Robert S. Barbieri[a], Brian Q. Rieksts[a], Kenneth L. Reitmeyer[b], Kenneth L. Stott Jr.[b]

[a]*Mathematics & CIS Department, Kutztown University, Kutztown, PA 19530, USA*
[b]*Homer Research Laboratories, Bethlehem Steel Corporation, Bethlehem, PA 18016, USA*

Received 1 October 1999; received in revised form 1 August 2000

Abstract

Let $G=(V,E)$ be a connected graph with specified vertex $v_0 \in V$, length $l(e) \geqslant 0$ for each $e \in E$, and positive parameters τ and γ. The cable-trench problem (CTP) is to find a spanning tree T such that $\tau l_\tau(T) + \gamma l_\gamma(T)$ is minimized where $l_\tau(T)$ is the total length of the spanning tree T and $l_\gamma(T)$ is the total path length in T from v_0 to all other vertices of V. Since all vertices must be connected to v_0 and only edges from E are allowed, the solution will not be a Steiner tree. Consider the ratio $R=\tau/\gamma$. For R large enough the solution will be a minimum spanning tree and for R small enough the solution will be a shortest path. In this paper, the CTP will be shown to be NP-complete. A mathematical formulation for the CTP will be provided for specific values of τ and γ. Also, a heuristic will be discussed that will solve the CTP for all values of R.

Scope and purpose

Both the shortest path and the minimum spanning tree problems are universally discussed in operations research and management science textbooks. Since the late 1950s, efficient algorithms have been known for both of these problems. In this paper, the cable-trench problem is defined which combines the shortest path problem and the minimum spanning tree problem to create a problem that is shown to be NP-complete. In other words, two easy problems are combined to get a more realistic problem that is difficult to solve. Examples are used to illustrate an efficient and effective heuristic solution procedure for the cable-trench problem.

Keywords: Minimum spanning tree; Shortest path; Networks; Graph theory; NP-completeness; Heuristics

* Corresponding author. Homer Research Laboratories, Bethlehem Steel Corporation, Bethlehem, PA 18016, USA. Tel.: 1-610–694-4595; fax: 1-610-694-2981.

E-mail address: vasko@kutztown.edu (F.J. Vasko).

0305-0548/01/$ - see front matter
PII: S0305-0548(00)00083-6

1. Introduction

For a connected graph $G = (V, E)$ with specified vertex $v_0 \in V$, let $l(e) \geqslant 0$ be the length of each $e \in E$. Also, let τ and γ denote positive parameters. The solution to the cable-trench problem (CTP) is the spanning tree T that minimizes $\tau l_\tau(T) + \gamma l_\gamma(T)$ where $l_\tau(T)$ is the total length of the spanning tree T and $l_\gamma(T)$ is the total path length in T from v_0 to all other vertices of V.

The name cable-trench comes from the fact that a physical application of this problem is the connection of buildings on a university campus with the building housing the main computer. The main computer building is v_0 and the buildings to be connected to the main computer building are the other vertices. The only allowable routes between two buildings for digging trenches and laying cables define the edges of the graph. Since only the edges that have endpoints that must be connected by cable to the main computer building are allowed as part of the solution spanning tree, the solution to the problem will not be a Steiner tree. A trench may carry more than one cable once it is dug and the trench cost is proportional to the total trench distance and the per unit cost of digging the trench is τ. The cable cost is proportional to the total length of the cable required and the per unit cost of cable is γ.

If $\gamma > 0$ and $\tau = 0$, then the solution to the CTP is any shortest path solution from v_0 to all other vertices of G. In contrast, if $\tau > 0$ and $\gamma = 0$, then the solution to the CTP is any minimum spanning tree. Thus, the solutions to the two limiting cases can be found efficiently. Namely, for $R = \tau/\gamma$ as $R \to 0$, the CTP solution is a spanning tree that is a shortest path solution from v_0 to all other vertices of G. As $R \to \infty$ the CTP solution is any minimum spanning tree for G.

In this paper, the general cable-trench problem (arbitrary positive values for τ and γ) will be shown to be NP-complete. Thus the combination of two easy-to-solve problems results in an intractable problem! A mathematical formulation for the CTP that can be used to generate optimum CTP solutions for any values of τ and γ will be provided. A heuristic solution procedure that will generate solution spanning trees for all positive values of R will also be discussed. After that, the generation of lower bounds and optimal solutions for the CTP, based on the heuristic solution, will be outlined. Several examples will be used to illustrate how well this simple heuristic performs. The next two sections will provide a literature review and solution properties of the CTP, respectively.

2. Background

Both the shortest path (SP) problem and the minimum spanning tree (MST) problem are the fundamental topics in courses dealing with graph theory [9], combinatorial optimization and integer programming [3,13,14,20], analysis of algorithms [4], discrete mathematics [16], and operations research and mathematical programming [10,18,19]. Since the late 1950s, efficient algorithms have been known for the minimum spanning tree problem [12,15] as well as for the shortest path spanning tree problem [5]. Chen and Zhang [2] define a constrained minimum spanning tree problem and provide a polynomial time ($O(n^2)$) algorithm for solving it. However, several constrained versions of the minimum spanning tree problem have been shown to be NP-complete [6,8,21].

Several researchers have looked at the minimum spanning tree and shortest path problems "simultaneously". Booth and Westbrook [1] developed an algorithm that can be used to perform edge cost sensitivity analysis for both the minimum spanning tree and shortest-path tree in a planar graph. Saltzman [17] provided a counterexample to the question "Is a MST also a SP tree for a determined vertex in a positive weighted connected graph?". Khuller et al. [11] gave an algorithm that finds a spanning tree in which the distance between any vertex and the root of the shortest-path tree is at most $1 + \mathrm{sqrt}(2)d$ times the shortest-path distance, and yet the total weight of the tree is at most $1 + \mathrm{sqrt}(2)/d$ times the weight of a minimum spanning tree where $d > 0$ is given. Eppstein [6] proved that finding the minimum spanning tree that minimizes the path length between a particular set of vertices is NP-complete.

3. CTP solution properties

In this section the solution to the CTP will be characterized as the ratio τ/γ varies from 0 to ∞. The results follow directly from Theorem 1.

Theorem 1. *Suppose T_1 and T_2 are two spanning trees such that $l_\tau(T_1)$ and $l_\tau(T_2)$ are the total edge lengths of T_1 and T_2, respectively, and $l_\gamma(T_1)$ and $l_\gamma(T_2)$ are the total path lengths from v_0 to all other vertices of V for T_1 and T_2, respectively. Also, assume that there exists an $\varepsilon > 0$, such that*

$$\tau l_\tau(T_2) + \gamma l_\gamma(T_2) < \tau l_\tau(T_1) + \gamma l_\gamma(T_1) \quad \textit{for all } \tau/\gamma > \varepsilon, \tag{3.1}$$

$$\tau l_\tau(T_1) + \gamma l_\gamma(T_1) < \tau l_\tau(T_2) + \gamma l_\gamma(T_2) \quad \textit{for all } 0 < \tau/\gamma < \varepsilon. \tag{3.2}$$

Then $l_\tau(T_2) < l_\tau(T_1)$ and $l_\gamma(T_2) > l_\gamma(T_1)$.

Proof. Eq. (3.1) implies that $l_\tau(T_2)(\tau/\gamma) + l_\gamma(T_2) < l_\tau(T_1)(\tau/\gamma) + l_\gamma(T_1)$ for all $\tau/\gamma > \varepsilon$. Which implies that $(l_\tau(T_2) - l_\tau(T_1))(\tau/\gamma) < l_\gamma(T_1) - l_\gamma(T_2)$ for all $\tau/\gamma > \varepsilon$. Since τ/γ can be made arbitrarily large, $l_\tau(T_2) - l_\tau(T_1) \leqslant 0$ or $l_\tau(T_2) \leqslant l_\tau(T_1)$. Eq. (3.2) implies that $l_\tau(T_1) + l_\gamma(T_1)(\gamma/\tau) < l_\tau(T_2) + l_\gamma(T_2)(\gamma/\tau)$ for all $\gamma/\tau > 1/\varepsilon$. This implies that $(l_\gamma(T_1) - l_\gamma(T_2))(\gamma/\tau) < l_\tau(T_2) - l_\tau(T_1)$ for all $\gamma/\tau > 1/\varepsilon$. Since γ/τ can be made arbitrarily large, $l_\gamma(T_1) - l_\gamma(T_2) \leqslant 0$ or $l_\gamma(T_1) \leqslant l_\gamma(T_2)$.

Assume $l_\tau(T_1) = l_\tau(T_2)$ and $l_\gamma(T_2) > l_\gamma(T_1)$, then $(l_\tau(T_2) - l_\tau(T_1))(\tau/\gamma) < l_\gamma(T_1) - l_\gamma(T_2)$ implies that $0 < l_\gamma(T_1) - l_\gamma(T_2) < 0$. A contradiction! Analogously, assume $l_\tau(T_2) < l_\tau(T_1)$ and $l_\gamma(T_2) = l_\gamma(T_1)$, then $(l_\gamma(T_1) - l_\gamma(T_2))(\gamma/\tau) < l_\tau(T_2) - l_\tau(T_1)$ implies that $0 < l_\tau(T_2) - l_\tau(T_1) < 0$. A contradiction! Finally, assume that $l_\tau(T_1) = l_\tau(T_2)$ and $l_\gamma(T_1) = l_\gamma(T_2)$ then both Eqs. (3.1) and (3.2) are contradicted! Therefore, $l_\tau(T_2) < l_\tau(T_1)$ and $l_\gamma(T_2) > l_\gamma(T_1)$.

Let T_1 be a spanning tree that is a shortest path solution from v_0 to all other vertices in V such that total edge length is minimized. Let $l_\tau(T_1)$ be the total trench length corresponding to T_1 and $l_\gamma(T_1)$ be the total cable length corresponding to T_1. Let T_Ω be a minimum spanning tree that minimizes the total path length from v_0 to all other vertices in V and let $l_\tau(T_\Omega)$ be the total trench length corresponding to T_Ω and $l_\gamma(T_\Omega)$ be the total cable length corresponding to T_Ω. If $l_\tau(T_1) = l_\tau(T_\Omega)$, then T_1 is the optimal spanning tree for all values of $\tau/\gamma > 0$. Otherwise, $l_\tau(T_1) > l_\tau(T_\Omega)$, $l_\gamma(T_1) < l_\gamma(T_\Omega)$, and $T_1 \neq T_\Omega$.

Assume $T_1 \neq T_\Omega$, then there exists a finite sequence of strictly increasing positive real numbers, i.e., $0 < \varepsilon_1 < \varepsilon_2 < \varepsilon_3 < \cdots < \varepsilon_N$ and corresponding spanning trees such that T_1 is the solution to the cable-trench problem for all $0 < \tau/\gamma \leqslant \varepsilon_1$, T_2 is the solution to the cable-trench problem for all $\varepsilon_1 \leqslant \tau/\gamma \leqslant \varepsilon_2$, T_3 is the solution to the cable-trench problem for all $\varepsilon_2 \leqslant \tau/\gamma \leqslant \varepsilon_3, \ldots, T_N$ is the solution to the cable-trench problem for all $\varepsilon_{N-1} \leqslant \tau/\gamma \leqslant \varepsilon_N$, and T_Ω is the solution to the cable-trench problem for all $\tau/\gamma \geqslant \varepsilon_N$.

Let $l_\tau(T_i)$ be the total trench length and $l_\gamma(T_i)$ be the total cable length corresponding to the spanning tree T_i, $(1 < i < \Omega)$ then the following inequalities must hold:

$$l_\tau(T_1) > l_\tau(T_2) > l_\tau(T_3) > \cdots > l_\tau(T_i) > \cdots > l_\tau(T_N) > l_\tau(T_\Omega)$$

and

$$l_\gamma(T_1) < l_\gamma(T_2) < l_\gamma(T_3) < \cdots < l_\gamma(T_i) < \cdots < l_\gamma(T_N) < l_\gamma(T_\Omega).$$

In other words, if $T_1 \neq T_\Omega$, then the solution to the CTP is a sequence of spanning trees such that as the τ/γ value increases, the total length of the spanning tree *strictly* decreases each time another spanning tree becomes optimum and the total path length from v_0 to all vertices of V *strictly* increases each time another spanning tree becomes optimum. This implies that when total cost versus τ/γ is plotted, the graph is a piecewise linear curve with *strictly decreasing* slopes.

A simple lower bound (SLB) for the CTP, for any $\tau/\gamma > 0$, is given by $\tau l_\tau(T_\Omega) + \gamma l_\gamma(T_1)$. A simple heuristic (hence an upper bound), denoted $T_1 - T_\Omega$, for the CTP is to use T_1 until T_Ω is lower in cost, then use T_Ω. Specifically, $\tau l_\tau(T_1) + \gamma l_\gamma(T_1) \leqslant \tau l_\tau(T_\Omega) + \gamma l_\gamma(T_\Omega)$ when $\tau/\gamma \leqslant (l_\gamma(T_\Omega) - l_\gamma(T_1))/(l_\tau(T_1) - l_\tau(T_\Omega))$ implies that T_1 should be used for $\tau/\gamma \leqslant (l_\gamma(T_\Omega) - l_\gamma(T_1))/(l_\tau(T_1) - l_\tau(T_\Omega))$ and T_Ω used for $\tau/\gamma \geqslant (l_\gamma(T_\Omega) - l_\gamma(T_1))/(l_\tau(T_1) - l_\tau(T_\Omega))$. If $T_1 - T_\Omega$ is used as a heuristic solution to the CTP, then its deviation from the optimal solution to the CTP is bounded by its deviation from SLB. The deviation of $T_1 - T_\Omega$ from SLB is

$$[(l_\tau(T_1) - l_\tau(T_\Omega))\tau/(l_\tau(T_\Omega)\tau + l_\gamma(T_1)\gamma)] \quad \text{for } \tau/\gamma \leqslant (l_\gamma(T_\Omega) - l_\gamma(T_1))/(l_\tau(T_1) - l_\tau(T_\Omega))$$

and

$$[(l_\gamma(T_\Omega) - l_\gamma(T_1))\gamma/(l_\tau(T_\Omega)\tau + l_\gamma(T_1)\gamma)] \quad \text{for } \tau/\gamma \geqslant (l_\gamma(T_\Omega) - l_\gamma(T_1))/(l_\tau(T_1) - l_\tau(T_\Omega)).$$

To illustrate the relationships discussed above, consider the following simple example.

Example 1. let $G = (V, E)$ where $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2),(1, 3),(2, 3),(2, 4),(3, 4)\}$ with edge lengths 16,12,8,7, and 4, respectively. Then it can be shown that the optimal CTP solution to this problem is:

$T_1 = \{(1, 2),(1, 3),(3, 4)\}$ for $0 < \tau/\gamma \leqslant 1/2$ with cost $32\tau + 44\gamma$,
$T_2 = \{(1, 3),(2, 3),(3, 4)\}$ for $1/2 \leqslant \tau/\gamma \leqslant 3$ with cost $24\tau + 48\gamma$,
and $T_\Omega = \{(1, 3),(2, 4),(3, 4)\}$ for $3 \leqslant \tau/\gamma$ with cost $23\tau + 51\gamma$.

This example shows that the intermediate (T_i for $1 < i < \Omega$) spanning tree solutions need not be contained in the union of T_1 and T_Ω, e.g. $T_2 = \{(1, 3),(2, 3),(3, 4)\} \not\subset \{(1, 2),(1, 3),(2, 4),(3, 4)\} = T_1 \cup T_\Omega$. An example that will illustrate that the intersection of T_1 and T_Ω need not be contained in each intermediate spanning tree solution will be provided later. In this particular example, adjacent (optimality intervals have common endpoints) optimal spanning trees differ by only one edge. However, as will be shown later, this is not true in general.

For this example, SLB $= 23\tau + 44\gamma$ for all $\tau/\gamma > 0$. The cost for the heuristic solution $T_1 - T_\Omega$ is $32\tau + 44\gamma$ for $\tau/\gamma \leqslant (51-44)/(32-23) = 7/9$ and the cost for $T_1 - T_\Omega$ is $23\tau + 51\gamma$ for $\tau/\gamma \geqslant 7/9$. The maximum deviation of $T_1 - T_\Omega$ from SLB occurs at 7/9 and is 11.3%. Later in this paper, a methodology for determining tighter lower bounds will be discussed.

4. Computational complexity of the CTP

Although, in the limiting cases, the CTP is either a shortest-path problem ($\gamma > 0$ and $\tau = 0$) or a minimum spanning tree problem ($\gamma = 0$ and $\tau > 0$), both of which are in P, in general, the CTP will be shown to be NP-complete.

Theorem 2. *Finding the minimum spanning tree that minimizes the path length between a particular set of vertices s and t is NP-complete.*

Proof. Eppstein provides a proof in Eppstein [6] by a reduction from 3SAT. □

Corollary 1. *Finding the minimum spanning tree that minimizes total path length from v_0 to all other vertices in V is NP-complete.*

Proof. Let $s = v_0$ and t be any vertex in $V - \{v_0\}$, then this problem is NP-complete because it is a more general case of the problem proven to be NP-complete in Theorem 2. □

Corollary 1 shows that, in general, finding the spanning tree T_Ω is NP-complete.

Corollary 2. *The cable-trench problem is NP-complete.*

Proof. The CTP is a more general case of the problem proven to be NP-complete in Corollary 1. □

Theorem 3. *Finding a spanning tree that is a shortest-path solution from v_0 to all other vertices in V such that total edge length is minimized is in P.*

Proof. Eppstein [7] provided the following proof. Find the shortest distance from v_0 to each vertex of $V - \{v_0\}$. Find the set of all directed edges that can be part of a shortest path, i.e., an edge (u, w) is in this set if $l(u, w) = \text{distance}(v_0, w) - \text{distance}(v_0, u)$. Find a minimum spanning arborescence of this directed graph. All the above steps can be carried out in polynomial time. □

Corollary 1. *Let $G = (V, E)$ be a connected graph with specified vertex $v_0 \in V$, length $l(e) = k \geqslant 0$ for each $e \in E$, i.e., all $e \in E$ have the same length, then the CTP for G can be solved for all $\tau/\gamma \geqslant 0$ in polynomial time.*

Proof. Since all edge lengths are equal, all spanning trees are minimal spanning trees. Therefore, $T_1 = T_\Omega$ and T_1 is the optimal spanning tree for the CTP for all $\tau/\gamma \geqslant 0$. □

In one extreme, when all the edge lengths of G are distinct, there is one unique minimum spanning tree for G (see Rosen [16, p. 570]). Therefore, T_Ω is the unique minimum spanning tree for G that is easy to generate. At the other extreme, when all the edge lengths are the same, $T_\Omega = T_1$ and is likewise easy to generate.

Since the CTP is NP-complete, a heuristic will be developed to solve it. This heuristic will take advantage of Theorem 3 that guarantees that T_1, the spanning tree that is a shortest path solution from v_0 to all other vertices in V such that total edge length is minimized, can be generated efficiently. Before discussing this heuristic, a mathematical formulation of the CTP will be given.

5. A mathematical formulation of the cable-trench problem

The following is a zero-one mixed integer linear programming formulation of the CTP (MFCTP).

$$\text{Minimize} \quad \gamma[\textstyle\sum\sum d_{ij}x_{ij}] + \tau[\sum\sum d_{ij}y_{ij}], \tag{5.1}$$

subject to

$$\textstyle\sum x_{1j} = n-1 \quad i = 1 \text{ (vertex1)}, \tag{5.2}$$

$$\textstyle\sum x_{ij} - \sum x_{ki} = -1 \quad i = 2, 3, \ldots, n \text{ (all other vertices)}, \tag{5.3}$$

$$\textstyle\sum y_{ij} = n-1 \quad \text{summed over all edges } (i<j), \tag{5.4}$$

$$(n-1)y_{ij} - x_{ij} - x_{ji} \geqslant 0 \quad i<j \text{ (for all edges)}, \tag{5.5}$$

$$x_{ij} \geqslant 0 \quad \text{for all } i,j, \tag{5.6}$$

$$y_{ij} = 0 \text{ or } 1 \quad \text{for all } i<j, \tag{5.7}$$

where, x_{ij} is the number of cables from vertex i to vertex j (no cables flow back to vertex 1 which is the computer center), y_{ij} is 1 if a trench is dug between vertex i and vertex j ($i<j$), 0 otherwise; and d_{ij} is the distance from vertex i to vertex j. The per unit cable cost is γ and the per unit trench cost is τ.

Constraint (5.2) ensures that $n-1$ cables leave the computer center. Constraint set (5.3) ensures that each of the $n-1$ buildings are connected by exactly one cable. Constraint (5.4) ensures that exactly $n-1$ trenches are dug. Although constraint (5.4) is not required to ensure that the optimal solution is found, it will be illustrated later that constraint (5.4) considerably strengthens this formulation by reducing the solution space that needs to be searched. Constraint set (5.5) ensures that cables are not laid unless a trench is dug. Although the number of cables (constraint set (5.6)) is only constrained to be nonnegative, these variables will, in fact, be integer because of their relationship to the trench variables which must be either zero or one (constraint set (5.7)).

To illustrate this formulation, the small graph discussed in Example 1 will be used. There are eight x_{ij} variables and five y_{ij} variables with five equality constraints (one per vertex and constraint (5.4)) and five inequality constraints (one per edge).

$$\text{Minimize} \quad \gamma[16x_{12} + 12x_{13} + 8x_{23} + 8x_{32} + 7x_{24} + 7x_{42} + 4x_{34} + 4x_{43}]$$

$$+ \ \tau[16y_{12} + 12y_{13} + 8y_{23} + 7y_{24} + 4y_{34}],$$

subject to

$$\begin{aligned}
&x_{12} + x_{13} = 3 \quad \text{vertex 1}\\
&x_{23} + x_{24} - x_{12} - x_{32} - x_{42} = -1 \quad \text{vertex 2}\\
&x_{32} + x_{34} - x_{13} - x_{23} - x_{43} = -1 \quad \text{vertex 3}\\
&x_{42} + x_{43} - x_{24} - x_{34} = -1 \quad \text{vertex 4}\\
&y_{12} + y_{13} + y_{23} + y_{24} + y_{34} = 3 \quad \text{constraint (5.4)}\\
&3y_{12} - x_{12} \geqslant 0 \quad \text{edge (1, 2)}\\
&3y_{13} - x_{13} \geqslant 0 \quad \text{edge (1, 3)}\\
&3y_{23} - x_{23} - x_{32} \geqslant 0 \quad \text{edge (2, 3)}\\
&3y_{24} - x_{24} - x_{42} \geqslant 0 \quad \text{edge (2, 4)}\\
&3y_{34} - x_{34} - x_{43} \geqslant 0 \quad \text{edge (3, 4)}\\
&x_{12}, x_{13}, x_{23}, x_{32}, x_{24}, x_{42}, x_{34}, x_{43} \geqslant 0,\\
&y_{12}, y_{13}, y_{23}, y_{24}, y_{34} \in \{0,\ 1\}.
\end{aligned}$$

6. A heuristic for the cable-trench problem

The heuristic outlined below starts by generating the spanning tree T_1 which can be accomplished efficiently as outlined in Theorem 3. Next, a one-opt neighborhood (edges interchanged one at a time) of this spanning tree is generated. Dominance checks are used to eliminate non-promising spanning trees from further consideration. Dominance checks are also used to determine if a generated spanning tree should replace a spanning tree currently in the solution sequence of spanning trees. Non-dominated spanning trees are candidates for the next spanning tree in the solution sequence. For each candidate spanning tree with cost $\tau l_\tau(T) + \gamma l_\gamma(T)$, the following inequality $\tau l_\tau(T) + \gamma l_\gamma(T) \leqslant \tau l_\tau(T_1) + \gamma l_\gamma(T_1)$ is solved for τ/γ and the candidate spanning tree with the *smallest* τ/γ value is chosen as the next spanning tree in the solution sequence. This procedure continues until the last spanning tree in the solution sequence is a minimum spanning tree.

Although this heuristic is a type of one-opt neighborhood search, the dominance rules allow for a kind of "backtracking" to occur so that spanning trees adjacent in the solution sequence may differ by *more* than one edge. Examples will be given later to illustrate this. The heuristic will now be outlined.

CTPHEUR1

Step 1 (Initialization): Use the procedure outlined in the proof of Theorem 3 to generate a spanning tree T_1 that is the shortest-path solution from v_0 to all other vertices in V such that total edge length is minimized. Generate a minimum spanning tree and denote its length as $l_\tau(T_\Omega)$. If $l_\tau(T_1) = l_\tau(T_\Omega)$, then T_1 is the optimal spanning tree for all values of $\tau/\gamma > 0$ and STOP. Otherwise, continue. Define arrays S, R, D, and L. Let T_1 be the first element of S. Assign the first element of R to be zero. The elements of S are sorted so that the following relationships hold:

$$l_\tau(T_1) > l_\tau(T_2) > l_\tau(T_3) > \cdots > l_\tau(T_{j-1}) > l_\tau(T_j) > \cdots$$

$$l_\gamma(T_1) < l_\gamma(T_2) < l_\gamma(T_3) < \cdots < l_\gamma(T_{j-1}) < l_\gamma(T_j) < \cdots .$$

Step 2: If T_q is the last spanning tree loaded into S, then sort the elements of S based on the relationships given in step 1. *Delete all* elements of S that come after T_q. Assume T_q is the jth solution in S and relabel it as T_j (T_j will denote the last spanning tree in the sorted list of spanning trees in S). Delete the elements of R (except $R_1 = 0$) and calculate $R_i = (l_\gamma(T_i) - l_\gamma(T_{i-1}))/(l_\tau(T_{i-1}) - l_\tau(T_i))$ $(i = 2, \ldots, j)$ as the breakeven ratio for spanning trees T_{i-1} and T_i.

If $R_i \leqslant R_{i-1}$, then delete spanning tree T_{i-1} from S and go to step 2.

Else ($R_i > R_{i-1}$) continue.

Let $E - T_j$ be the edges in E that are not in T_j. Let $D = \varnothing$ and $L = \varnothing$.

Step 3: Remove an edge from $E - T_j$ and insert it into T_j to form a loop. Insert all the edges of T_j that are in this loop into L.

Step 4: Remove an edge from L, call the resulting spanning tree T_p. Compute $l_\tau(T_p)$ and $l_\gamma(T_p)$. Consider the following cases:

Case 1: If $l_\tau(T_p) \leqslant l_\tau(T_s)$ and $l_\gamma(T_p) < l_\gamma(T_s)$ or if $l_\tau(T_p) < l_\tau(T_s)$ and $l_\gamma(T_p) \leqslant l_\gamma(T_s)$ for any $s \in S$, then delete T_s from S and insert T_p as the last element in S. Go to step 2. (*T_p dominates T_s*)

Case 2: If $l_\tau(T_p) \geqslant l_\tau(T_s)$ and $l_\gamma(T_p) > l_\gamma(T_s)$ or if $l_\tau(T_p) > l_\tau(T_s)$ and $l_\gamma(T_p) \geqslant l_\gamma(T_s)$ for any $s \in S$, then go to step 5. (*T_p is dominated by T_s — do not put in D*)

Case 3: If $l_\tau(T_p) > l_\tau(T_j)$ and $l_\gamma(T_p) < l_\gamma(T_j)$, then compute $\tau/\gamma = (l_\gamma(T_p) - l_\gamma(T_j))/(l_\tau(T_j) - l_\tau(T_p))$. If $\tau/\gamma > R_j$, then delete T_j from S and insert T_p as the last element in S. Go to step 2. (*T_p dominates T_j*)

Else ($\tau/\gamma \leqslant R_j$) go to step 5. (*$T_p$ is dominated — do not put in D*)

Case 4: If $l_\tau(T_p) = l_\tau(T_s)$ and $l_\gamma(T_p) = l_\gamma(T_s)$ for any $s \in S$, then compare T_p to T_s using the equal-cost components tie-breaking rule given below.

If T_s is chosen, then go to step 5.

If T_p is chosen, then delete T_s from S and insert T_p as the last element in S. Go to step 2. (*If T_p is "equal" to any element of S check tie-breaking rule between T_p and T_s.*)

Case 5: Otherwise, add T_p to D. Go to step 5.

Step 5: If $L \neq \varnothing$, then go to step 4,

else, if $E - T_j \neq \varnothing$, go to step 3, else go to step 6.

Step 6: If $D = \varnothing$ or $l_\tau(T_j) = l_\tau(T_\Omega)$ then STOP and print solutions and ratio information.

Otherwise, for each element T_i in D,
compute $\tau/\gamma = (l_\gamma(T_i) - l_\gamma(T_j))/(l_\tau(T_j) - l_\tau(T_i))$.

Step 7: Select the spanning tree in D with the *lowest* τ/γ value (in case of ties, see the tie-breaking rules given below) and denote this spanning tree as T_L. Insert T_L as the last element in S. Go to step 2.

S is an array of solution spanning trees and R is an array of τ/γ ratios at which the lowest cost solution changes. The array D is used to store candidate spanning trees and L is an array used to store edges that need to be removed one-at-a-time from a loop created by inserting an edge into a spanning tree.

Tie-breaking rules

(*A*) *Same ratio only*: If two spanning trees, T_i and $T_{i'}$ have the same ratio when compared to T_j, but $l_\tau(T_i) \neq l_\tau(T_{i'})$ and $l_\gamma(T_i) \neq l_\gamma(T_{i'})$, then calculate the following:

$$\mathrm{TB}_i = |l_\tau(T_i) - l_\tau(T_j)| + |l_\gamma(T_i) - l_\gamma(T_j)|$$

$$\mathrm{TB}_{i'} = |l_\tau(T_{i'}) - l_\tau(T_j)| + |l_\gamma(T_{i'}) - l_\gamma(T_j)|$$

If $\mathrm{TB}_i > \mathrm{TB}_{i'}$, then choose $T_{i'}$, otherwise choose T_i.

(*B*) *Same ratio and* $l_\tau(T_i) = l_\tau(T_{i'})$ *and* $l_\gamma(T_i) = l_\gamma(T_{i'})$: In this case, calculate the following for each spanning tree.

$$\mathrm{LS}_i = \Sigma(\#\text{of edges in } T_i \text{ from } v_0 \text{ to vertex } i)(\text{degree } i)$$

$$LS_{i'} = \Sigma(\#\text{of edges in } T_{i'} \text{ from } v_0 \text{ to vertex } i)(\text{degree } i)$$

The summations above are over all vertices except v_0.

If $\mathrm{LS}_i < \mathrm{LS}_i'$, then choose T_i, otherwise choose $T_{i'}$.

LS is a measure of loop size or number of candidates generated from a one-opt search — the smaller the value of LS, the larger the loops created when an edge is inserted into the spanning tree.

To illustrate this heuristic, the solution of the 4-vertex example discussed earlier, i.e., $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 4)\}$ with edge lengths 16, 12, 8, 7, and 4 respectively, will now be outlined.

For this graph, T_1 is determined to be $\{(1, 2), (1, 3), (3, 4)\}$ with cost $32\tau + 44\gamma$ and the minimum spanning tree length is determined to be $l_\tau(T_\Omega) = 23$.

$E - T_1 = \{(2, 3), (2, 4)\}$ and the neighbors of T_1 are given in Table 1.

The spanning tree with $\tau/\gamma = \frac{1}{2}$ is chosen and inserted into S, the sorted collection of spanning tree solutions (Table 2).

$E - T_2 = \{(1, 2), (2, 4)\}$ and the neighbors of T_2 are given in Table 3.

The spanning tree with $\tau/\gamma = 3$ is chosen (there is only one candidate) and inserted into S (Table 4).

Since the spanning tree length of the last spanning tree in S is equal to $l_\tau(T_\Omega)$, the heuristic terminates with the CTP solution of:

$T_1 = \{(1, 2), (1, 3), (3, 4)\}$ for $0 < \tau/\gamma \leqslant 1/2$ with cost $32\tau + 44\gamma$,
$T_2 = \{(1, 3), (2, 3), (3, 4)\}$ for $1/2 \leqslant \tau/\gamma \leqslant 3$ with cost $24\tau + 48\gamma$,
and $T_\Omega = \{(1, 3), (2, 4), (3, 4)\}$ for $3 \leqslant \tau/\gamma$ with cost $23\tau + 51\gamma$.

Because this heuristic generates solutions such that the graph of total cost versus τ/γ is a piecewise linear curve with strictly decreasing slopes, the same as the optimal solution (discussed earlier), only the breakpoints need to be checked for optimality. By solving the MFCTP at the breakpoints, i.e., $\tau/\gamma = 1/2$ and 3, the optimality of this heuristic solution was verified.

Table 1

Edge in	Edge out	Spanning tree cost	Status	Step7 τ/γ value
(2, 3)	(1, 2)	$24\tau + 48\gamma$	Candidate	1/2
(2, 3)	(1, 3)	$28\tau + 68\gamma$	Candidate	6
(2, 4)	(1, 2)	$23\tau + 51\gamma$	Candidate	7/9
(2, 4)	(1, 3)	$27\tau + 66\gamma$	Candidate	22/5
(2, 4)	(3, 4)	$35\tau + 51\gamma$	Dominated	NA

Table 2

S			*R*
Spanning tree	Spanning tree length	Total path length from v_0 to all other vertices of V	
(1, 2), (1, 3), (3, 4)	32	44	0
(1, 3), (2, 3), (3, 4)	24	48	1/2

Table 3

Edge in	Edge out	Spanning tree cost	Status	Step7 τ/γ value
(1, 2)	(1, 3)	$28\tau + 68\gamma$	Dominated	NA
(1, 2)	(2, 3)	$32\tau + 44\gamma$	Dominated	NA
(2, 4)	(2, 3)	$23\tau + 51\gamma$	Candidate	3
(2, 4)	(3, 4)	$27\tau + 59\gamma$	Dominated	NA

Table 4

S			*R*
Spanning tree	Spanning tree length	Total path length from v_0 to all other vertices of V	
(1, 2), (1, 3), (3, 4)	32	44	0
(1, 3), (2, 3), (3, 4)	24	48	1/2
(1, 3), (2, 4), (3, 4)	23	51	3

This heuristic, which solves the CTP for all $\tau/\gamma > 0$, was coded in C + + and several examples (up to 20 nodes and 39 arcs) were solved on a 500 MHz Pentium PC requiring less than 1 s of execution time for each problem.

7. Lower bounds and optimal solutions to the CTP

Given a connected graph, determine the CTPHEUR1 solution to the CTP for this graph. If $T_1 = T_\Omega$, then T_1 is the optimal solution to the CTP for all $\tau/\gamma > 0$.

If $T_1 \neq T_\Omega$, then there exists a finite sequence of strictly increasing positive real numbers, i.e., $0 < \lambda_1 < \lambda_2 < \lambda_3 < \cdots < \lambda_N$ at which the CTPHEUR1 solution changes spanning trees. In order to find either a tighter lower bound (than SLB), an improved heuristic solution, or the optimal CTP solution for all $\tau/\gamma > 0$, we need to focus on the "breakpoints" i.e., the $\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_N$ at which the CTPHEUR1 solution changes spanning trees. Specifically, alternate between using the math formulation (MFCTP) given earlier combined with optimization software to find the optimal solution to the CTP at one or more of the breakpoints, and, if the heuristic solutions at the breakpoints are not optimal, then use the optimal solutions to find new breakpoints. An example will now be provided to illustrate this methodology.

Example 2. The CTP solution of the graph given by $V = \{1, 2, 3, 4, 5, 6, 7\}$ and $E = \{(1, 2), (1, 3), (1, 7), (2, 4), (3, 4), (3, 5), (3, 6), (3, 7), (4, 5), (5, 6), (6, 7)\}$ with edge lengths 50, 60, 60, 30, 30, 40, 40, 10, 30, 30, and 39, respectively, will now be given.

The CTP solution generated for this problem by CTPHEUR1 is:

$T_1 = \{(1, 2), (1, 3)(1, 7), (2, 4), (3, 5), (6, 7)\}$ for $0 \leqslant \tau/\gamma \leqslant 11/49$ with cost $279\tau + 449\gamma$,

$T_2 = \{(1, 2), (1, 3), (2, 4), (3, 5), (3, 6), (3, 7)\}$ for $11/49 \leqslant \tau/\gamma \leqslant 1$ with cost $230\tau + 460\gamma$,

$T_3 = \{(1, 2), (1, 3), (2, 4), (3, 6), (3, 7), (4, 5)\}$ for $1 \leqslant \tau/\gamma \leqslant 7/2$ with cost $220\tau + 470\gamma$,

and $T_\Omega = \{(1, 2), (2, 4), (3, 4), (3, 7), (4, 5), (5, 6)\}$ for $7/2 \leqslant \tau/\gamma$ with cost $180\tau + 610\gamma$.

For this problem, SLB = $180\tau + 449\gamma$ and the CTPHEUR1 deviations from SLB at the three breakpoints (11/49, 1 and 7/2) are 4.5, 9.7, and 14.9%, respectively, with the maximum deviation from SLB for all $\tau/\gamma > 0$ of 14.9%. To determine an improved lower bound (ILB1), the MFCTP is solved using Frontline Systems Premium Solver Plus (Version 3.5) for Microsoft Excel at the two extreme breakpoints, 11/49 and 7/2. The optimal solution at 11/49 agreed with the CTPHEUR1 solution, but the optimal solution at 7/2 was strictly better than the CTPHEUR1 solution. The cost function for the optimal spanning tree at 7/2 was compared to the cost function for T_Ω to determine a new breakpoint of 141/39 that was then proved optimal by solving the MFCTP at this point.

An improved lower bound (ILB1) consists of the T_1 cost function for $\tau/\gamma \leqslant 11/49$, $220.95\tau + 462.03\gamma$ for $11/49 \leqslant \tau/\gamma \leqslant 141/39$, and the T_Ω cost function for $\tau/\gamma \leqslant 141/39$. The lower bound on the cost for $11/49 \leqslant \tau/\gamma \leqslant 141/39$ is simply the line segment connecting the T_1 cost function at 11/49 with the T_Ω cost function at 141/39. This improved lower bound indicates that the CTPHEUR1 solution is optimal for $\tau/\gamma \leqslant 11/49$ and $\tau/\gamma \geqslant 141/39$ and deviates from the lower bound by 1.03% at $\tau/\gamma = 1$ and by 0.38% at $\tau/\gamma = 7/2$ — a maximum deviation of 1.03%. This is a considerable improvement over SLB and required only a few instances of the MFCTP to be solved.

In order to use the CTPHEUR1 solution to generate either even tighter lower bounds or the optimal CTP solution for all $\tau/\gamma > 0$, this procedure simply needs to be extended to more breakpoints. For a CTPHEUR1 solution with many breakpoints, the next breakpoint evaluated should be approximately in the middle, i.e., $N/2$, if N is even. In this example the only remaining CTPHEUR1 breakpoint is 1. Solving the MFCTP at this point indicates that the CTPHEUR1 solution is not optimal for $\tau/\gamma = 1$. Using this optimal solution, a new breakpoint of 9/11 is found and proven optimum. Therefore, the CTPHEUR1 solution combined with only five solutions of the MFCTP determined the optimal solution for the CTP for all $\tau/\gamma > 0$.

The optimal CTP solution for Example 2 is:

$T_1 = \{(1, 2), (1, 3)(1, 7), (2, 4), (3, 5), (6, 7)\}$ for $0 \leqslant \tau/\gamma \leqslant 11/49$ with cost $279\tau + 449\gamma$,

$T_2 = \{(1, 2), (1, 3), (2, 4), (3, 5), (3, 6), (3, 7)\}$ for $11/49 \leqslant \tau/\gamma \leqslant 9/11$ with cost $230\tau + 460\gamma$,

$T_3 = \{(1, 2), (1, 3), (2, 4), (3, 7), (4, 5), (6, 7)\}$ for $9/11 \leqslant \tau/\gamma \leqslant 141/394$ with cost $219\tau + 469\gamma$,

and $T_\Omega = \{(1, 2), (2, 4), (3, 4), (3, 7), (4, 5), (5, 6)\}$ for $141/39 \leqslant \tau/\gamma$ with cost $180\tau + 610\gamma$.

For $0 \leqslant \tau/\gamma \leqslant 9/11$ and for $\tau/\gamma \geqslant 141/39$, the CTPHEUR1 solution is optimum. For $9/11 \leqslant \tau/\gamma \leqslant 1$, the CTPHEUR1 solution deviates from the optimum by $[230(\tau/\gamma) + 460]/[219(\tau/\gamma) + 469] - 1$ with a maximum deviation of 0.2%. For $1 \leqslant \tau/\gamma \leqslant 7/2$, the CTPHEUR1 solution deviates from the optimum by $[220(\tau/\gamma) + 470]/[219(\tau/\gamma) + 469] - 1$ with a maximum deviation of 0.36%. For $7/2 \leqslant \tau/\gamma \leqslant 141/39$, the CTPHEUR1 solution deviates from the optimum by $[180(\tau/\gamma) + 610]/[219(\tau/\gamma) + 469] - 1$ with a maximum deviation of 0.36%.

As seen in this example (and can be shown to be true in general), if the CTPHEUR1 solution is not optimum, then the maximum deviation of the CTPHEUR1 solution from the optimum solution will occur at the breakpoints of the CTPHEUR1.

8. Additional observations

In this section, several examples will be used to further illustrate the properties of the CTP. Details of these examples are given in the appendix. The complete CTPHEUR1 solution to example 3 is given to illustrate the "backtracking" nature of this heuristic. Observe that T_2 and T_Ω differ by two edges. The heuristic was able to identify and delete a spanning tree in S in order to improve the CTP solution. By solving the MFCTP at the breakpoints $\tau/\gamma = 1$ and 10/3, the optimality of this heuristic solution was verified.

Since $T_1 \cap T_\Omega \not\subset T_4$, example 4 illustrates that the intersection of T_1 and T_Ω need not be contained within each solution spanning tree. Again, by solving the MFCTP at the breakpoints $\tau/\gamma = 1/4$, 1, 7, and 28, the optimality of this heuristic solution was verified.

Example 5 is a somewhat larger problem with 20 vertices and 39 edges. The CTPHEUR1 solution consists of six spanning trees and five breakpoints. In this example T_1 and T_2 differ by 3 edges, T_2 and T_3 differ by 1 edge, T_3 and T_4 differ by 3 edges, T_4 and T_5 differ by 1 edge, and T_5 and T_Ω differ by 2 edges. Although several adjacent spanning trees of the CTPHEUR1 solution differ by as many as three edges, indicating significant backtracking occurred, this solution was not optimum.

The optimal solution to example 5 is given in the appendix and consists of five spanning trees and four breakpoints. Deviation of the CTPHEUR1 solution from the optimal solution occurs for $4.8 \leqslant \tau/\gamma \leqslant 9.5$ with a maximum deviation of 0.23%. The reason CTPHEUR1, basically a somewhat sophisticated one-opt neighborhood search, was unable to find the optimal solution for example 5 was that, in the optimal solution, T_3 and T_4 as well as T_4 and T_Ω differ by *five* edges.

In general, the backtracking capability of CTPHEUR1 allows it to find adjacent spanning trees that differ by more than one edge — only if these spanning trees can be generated via a sequence of strictly improving spanning trees (this is illustrated by the detailed CTPHEUR1 solution of example 1 given in the Appendix). The more the optimal solution contains adjacent spanning trees that differ by several edges, the less likely that the CTPHEUR1 solution will be optimal for all $\tau/\gamma > 0$. However, even if the CTPHEUR1 solution is not optimal for all $\tau/\gamma > 0$, the methodology discussed in the previous section allows for the efficient generation of either an improved heuristic solution or the optimal CTP solution for all $\tau/\gamma > 0$.

Finally, example 5 illustrates the significant computational benefit of including constraint (5.4) in the MFCTP. For any specific τ/γ value, examples 1–4 solved in a matter of seconds on a 500 MHz Pentium PC — with or without constraint (5.4). However, for example 5, for several of the CTPHEUR1 as well as optimal solution breakpoints, solving the MFCTP (for each breakpoint) without constraint (5.4) required *more than 10 h* of computer time. With the addition of constraint (5.4) to the formulation, using the same computer, optimization software, and software parameter settings, the same problems were solved in *less than* 2 *min*! What a difference one constraint can make at strengthening a problem formulation!

9. Conclusions

Two easy-to-solve graph theoretic problems, the minimal spanning tree problem and the problem of finding the shortest path from a given vertex to all other vertices of a graph have been combined to create, in general, an NP-complete problem called the cable-trench problem. Although the cable-trench problem is essentially a composite of two problems each of which can be solved very efficiently in polynomial time, its NP-completeness follows easily from known graph theoretic results. A mathematical optimization formulation that can be used to determine the optimum CTP solution for any $\tau/\gamma > 0$ is provided. The heuristic developed for the cable-trench problem, basically a neighborhood search procedure, tends to generate either optimum or near-optimum results very quickly for all $\tau/\gamma > 0$. It was also shown as to how the heuristic solution, combined with optimal solutions for a few problem-specific τ/γ values, could be used to generate both tight lower bounds and the optimal CTP solution for all $\tau/\gamma > 0$.

Appendix

Example 3. The CTP solution of the graph given by $V = \{1, 2, 3, 4, 5, 6\}$ and $E = \{(1, 2), (1, 3), (2, 4), (2, 5), (2, 6), (3, 4), (4, 5), (5, 6)\}$ with edge lengths 6, 5, 4, 4, 4, 3, 3, and 3, respectively, will now be outlined.

For this graph, T_1 is determined to be $\{(1, 2), (1, 3), (2, 5), (2, 6), (3, 4)\}$ with cost $22\gamma + 39\gamma$ and the minimum spanning tree length is determined to be $l_\tau(T_\Omega) = 18$. $E - T_1 = \{(2, 4), (4, 5), (5, 6)\}$ and the neighbors of T_1 are given in Table 5.

Table 5

Edge in	Edge out	Spanning tree cost	Status	Step7 τ/γ value
(2, 4)	(1, 2)	$20\tau + 57\gamma$	Candidate	9
(2, 4)	(1, 3)	$21\tau + 49\gamma$	Candidate	10
(2, 4)	(3, 4)	$23\tau + 41\gamma$	Dominated	NA
(4, 5)	(1, 2)	$19\tau + 58\gamma$	Candidate	19/3
(4, 5)	(1, 3)	$20\tau + 55\gamma$	Candidate	8
(4, 5)	(2, 5)	$21\tau + 40\gamma$	Candidate	1
(4, 5)	(3, 4)	$22\tau + 44\gamma$	Dominated	NA
(5, 6)	(2, 5)	$21\tau + 42\gamma$	Candidate	3
(5, 6)	(2, 6)	$21\tau + 42\gamma$	Candidate	3

Table 6

S			R
Spanning tree	Spanning tree length	Total path length from v_0 to all other vertices of V	
(1, 2), (1, 3), (2, 5), (2, 6), (3, 4)	22	39	0
(1, 2), (1, 3), (2, 6), (3, 4), (4, 5)	21	40	1

Table 7

Edge in	Edge out	Spanning tree cost	Status	Step7 τ/γ value
(2, 4)	(1, 2)	$19\tau + 52\gamma$	Candidate	6
(2, 4)	(1, 3)	$20\tau + 52\gamma$	Candidate	12
(2, 4)	(3, 4)	$22\tau + 44\gamma$	Dominated	NA
(2, 5)	(1, 2)	$19\tau + 58\gamma$	Candidate	9
(2, 5)	(1, 3)	$20\tau + 55\gamma$	Candidate	15
(2, 5)	(3, 4)	$22\tau\mathrm{T} + 44\gamma$	Dominated	NA
(2, 5)	(4, 5)	$22\tau + 39\gamma$	Dominated	NA
(5, 6)	(1, 2)	$18\tau + 56\gamma$	Candidate	16/3
(5, 6)	(1, 3)	$19\tau + 64\gamma$	Candidate	12
(5, 6)	(2, 6)	$20\tau + 44\gamma$	Candidate	4
(5, 6)	(3, 4)	$21\tau + 50\gamma$	Dominated	NA
(5, 6)	(4, 5)	$21\tau + 42\gamma$	Dominated	NA

The spanning tree with $\tau/\gamma = 1$ is chosen and inserted into S (Table 6).
$E - T_2 = \{(2, 4), (2, 5), (5, 6)\}$ and the neighbors of T_2 are given in Table 7.
The spanning tree with $\tau/\gamma = 4$ is chosen and inserted into S (Table 8).

Table 8

S			R
Spanning tree	Spanning tree length	Total path length from v_0 to all other vertices of V	
(1, 2), (1, 3), (2, 5), (2, 6), (3, 4)	22	39	0
(1, 2), (1, 3), (2, 6), (3, 4), (4, 5)	21	40	1
(1, 2), (1, 3)(3, 4), (4, 5), (5, 6)	20	44	4

Table 9

Edge in	Edge out	Spanning tree cost	Status	Step7 τ/γ value
(2, 4)	(1, 2)	$18\tau + 50\gamma$	Candidate	3
(2, 4)	(1, 3)	$19\tau + 58\gamma$	Candidate	14
(2, 4)	(3, 4)	$21\tau + 50\gamma$	Dominated	NA
(2, 5)	(1, 2)	$18\tau + 53\gamma$	Candidate	4.5
(2, 5)	(1, 3)	$19\tau + 58\gamma$	Candidate	14
(2, 5)	(3, 4)	$21\tau + 47\gamma$	Dominated	NA
(2, 5)	(4, 5)	$21\tau + 42\gamma$	Dominated	NA
(2, 6)	(1, 2)	$18\tau + 56\gamma$	Candidate	6
(2, 6)	(1, 3)	$19\tau + 64\gamma$	Candidate	20
(2, 6)	(3, 4)	$21\tau + 50\gamma$	Dominated	NA
(2, 6)	(4, 5)	$21\tau + 42\gamma$	Dominated	NA
(2, 6)	(5, 6)	$21\tau + 40\gamma$	Dominated	NA

Table 10

S			R
Spanning tree	Spanning tree length	Total path length from v_0 to all other vertices of V	
(1, 2), (1, 3), (2, 5), (2, 6), (3, 4)	22	39	0
(1, 2), (1, 3), (2, 6), (3, 4), (4, 5)	21	40	1
(1, 2), (1, 3)(3, 4), (4, 5), (5, 6)	20	44	4
(1, 3), (2, 4), (3, 4), (4, 5), (5, 6)	18	50	3

$E - T_3 = \{(2, 4), (2, 5), (2, 6)\}$ and the neighbors of T_3 are given in Table 9.
The spanning tree with $\tau/\gamma = 3$ is chosen and inserted into S (Table 10).
Since $R_3 > R_4, T_3$ is deleted from S and the ratios are recalculated (Table 11).

Table 11

S			R
Spanning tree	Spanning tree length	Total path length from v_0 to all other vertices of V	
(1, 2), (1, 3), (2, 5), (2, 6), (3, 4)	22	39	0
(1, 2), (1, 3), (2, 6), (3, 4), (4, 5)	21	40	1
(1, 3), (2, 4), (3, 4), (4, 5), (5, 6)	18	50	10/3

Since the spanning tree length of the last spanning tree in S is equal to $l_\tau(T_\Omega)$, the heuristic terminates with the CTP solution of:
$T_1 = \{(1, 2), (1, 3), (2, 5), (2, 6), (3, 4)\}$ for $0 \leqslant \tau/\gamma \leqslant 1$ with cost $22\tau + 39\gamma$,
$T_2 = \{(1, 2), (1, 3), (2, 6), (3, 4), (4, 5)\}$ for $1 \leqslant \tau/\gamma \leqslant 10/3$ with cost $21\tau + 40\gamma$,
and $T_\Omega = \{(1, 3), (2, 4), (3, 4), (4, 5), (5, 6)\}$ for $10/3 \leqslant \tau/\gamma$ with cost $18\tau + 50\gamma$.

Example 4. Let $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $E = \{(1, 2), (1, 3), (1, 4), (2, 4), (2, 6), (3, 4), (3, 5), (4, 5), (4, 6), (4, 7), (5, 7), (5, 8), (6, 7), (6, 9), (7, 9), (8, 9)\}$ with edge lengths 7, 10, 8, 6, 7, 4, 3, 6, 5, 8, 6, 8, 4, 7, 5, and 9, respectively. The CTP solution generated for this problem by CTPHEUR1 is:
$T_1 = \{(1, 2), (1, 3), (1, 4), (3, 5), (4, 6), (4, 7), (5, 8), (6, 9)\}$ for $0 \leqslant \tau/\gamma \leqslant 1/4$ with cost $56\tau + 108\gamma$,
$T_2 = \{(1, 2), (1, 3), (1, 4), (3, 5), (4, 6), (5, 8), (6, 7), (6, 9)\}$ for $1/4 \leqslant \tau/\gamma \leqslant 1$ with cost $52\tau + 109\gamma$,
$T_3 = \{(1, 2), (1, 4), (3, 4), (3, 5), (4, 6), (5, 8), (6, 7), (7, 9)\}$ for $1 \leqslant \tau/\gamma \leqslant 7$ with cost $44\tau + 117\gamma$,
$T_4 = \{(1, 4), (2, 4), (3, 4), (3, 5), (4, 6), (5, 8), (6, 7), (7, 9)\}$ for $7 \leqslant \tau/\gamma \leqslant 28$ with cost $43\tau + 124\gamma$,
and $T_\Omega = \{(1, 2), (2, 4), (3, 4), (3, 5), (4, 6), (5, 8), (6, 7),(7, 9)\}$ for $28 \leqslant \tau/\gamma$ with cost $42\tau + 152\gamma$.

Example 5. Let $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$ AND $E = \{(1, 2), (1, 8), (2, 3), (2, 9), (2, 11), (3, 4), (4, 5), (4, 11), (4, 13), (5, 6), (6, 7), (6, 13), (6, 15), (7, 8), (8, 9), (8, 15), (9, 10), (9, 16), (10, 11), (10, 17), (10, 18), (11, 12), (12, 13), (12, 18), (12, 19), (13, 14), (14, 15), (14, 19), (14, 20), (15, 16), (16, 17), (16, 20), (17, 18), (17, 20), (18, 19), (19, 20)\}$ with edge lengths 10, 10, 10, 6, 6, 10, 10, 6, 6, 10, 10, 6, 6, 10, 6, 6, 3, 3, 3, 1, 1, 3, 3, 1, 1, 3, 3, 1, 1, 3, 1, 1, 3, 3, 3, and 3, respectively.

The CTP solution generated for this problem by CTPHEUR1 is:

$T_1 = \{(1, 2), (1, 8), (2, 3), (2, 9), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 15), (9, 10), (9, 16), (10, 17), (10, 18), (11, 12), (12, 13), (12, 19), (14, 15), (14, 20)\}$ for $0 \leqslant \tau/\gamma \leqslant 1$ with cost $99\tau + 362\gamma$,

$T_2 = \{(1, 2), (1, 8), (2, 3), (2, 9), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 15), (9, 16), (10, 17), (11, 12), (12, 13), (12, 18), (12, 19), (14, 20), (16, 17), (16, 20)\}$ for $1 \leqslant \tau/\gamma \leqslant 2$ with cost $95\tau + 366\gamma$,

$T_3 = \{(1, 2), (1, 8), (2, 3), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 15), (9, 16), (10, 17), (11, 12), (12, 13), (12, 18), (12, 19), (14, 20), (15, 16), (16, 17), (16, 20)\}$ for $2 \leqslant \tau/\gamma \leqslant 5$ with cost $92\tau + 372\gamma$,

$T_4 = \{(1, 2), (1, 8), (2, 3), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 15), (9, 10), (10, 11), (10, 17), (10, 18), (12, 13), (12, 18), (12, 19), (14, 20), (16, 17), (16, 20)\}$ for $5 \leqslant \tau/\gamma \leqslant 16/3$ with cost $90\tau + 382\gamma$,

$T_5 = \{(1, 2), (1, 8), (2, 3), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (9, 10), (10, 11), (10, 17), (10, 18), (12, 13), (12, 18), (12, 19), (14, 20), (15, 16), (16, 17), (16, 20)\}$ for $16/3 \leqslant \tau/\gamma \leqslant 9$ with cost $87\tau + 398\gamma$,

and $T_\Omega = \{(1, 2), (2, 3), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 9), (9, 10), (10, 11), (10, 17), (10, 18), (12, 13), (12, 18), (12, 19), (14,20), (15, 16), (16, 17), (16, 20)\}$ for $9 \leqslant \tau/\gamma$ with cost $83\tau + 434\gamma$.

The optimal solution for this problem is:

$T_1 = \{(1, 2), (1, 8), (2, 3), (2, 9), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 15), (9, 10), (9, 16), (10, 17), (10, 18), (11, 12), (12, 13), (12, 19), (14, 15), (14, 20)\}$ for $0 \leqslant \tau/\gamma \leqslant 1$ with cost $99\tau + 362\gamma$,

$T_2 = \{(1, 2), (1, 8), (2, 3), (2, 9), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 15), (9, 16), (10, 17), (11, 12), (12, 13), (12, 18), (12, 19), (14, 20), (16, 17), (16, 20)\}$ for $1 \leqslant \tau/\gamma \leqslant 2$ with cost $95\tau + 366\gamma$,

$T_3 = \{(1, 2), (1, 8), (2, 3), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 15), (9, 16), (10, 17), (11, 12), (12, 13), (12, 18), (12, 19), (14, 20), (15, 16), (16, 17), (16, 20)\}$ for $2 \leqslant \tau/\gamma \leqslant 4.8$ with cost $92\tau + 372\gamma$,

$T_4 = \{(1, 2), (1, 8), (2, 3), (2, 11), (4, 5), (4, 11), (6, 13), (7, 8), (9, 10), (10, 17), (10, 18), (11, 12), (12, 13), (12, 18), (12, 19), (14, 15), (14, 19), (14, 20), (16, 20)\}$ for $4.8 \leqslant \tau/\gamma \leqslant 9.5$ with cost $87\tau + 396\gamma$,

and $T_\Omega = \{(1, 2), (2, 3), (2, 11), (4, 5), (4, 11), (6, 15), (7, 8), (8, 9), (9, 10), (10, 11), (10, 17), (10, 18), (12, 13), (12, 18), (12, 19), (14, 20), (15, 16), (16, 17), (16, 20)\}$ for $9.5 \leqslant \tau/\gamma$ with cost $83\tau + 434\gamma$.

For $0 \leqslant \tau/\gamma \leqslant 4.8$ and for $\tau/\gamma \geqslant 9.5$, the CTPHEUR1 solution is optimum.

For $4.8 \leqslant \tau/\gamma \leqslant 5$, the CTPHEUR1 solution deviates from the optimum by $[92(\tau/\gamma) + 372]/[87(\tau/\gamma) + 396] - 1$ with a maximum deviation of 0.12%.

For $5 \leqslant \tau/\gamma \leqslant 16/3$, the CTPHEUR1 solution deviates from the optimum by $[90(\tau/\gamma) + 382]/[87(\tau/\gamma) + 396] - 1$ with a maximum deviation of 0.23%.

For $16/3 \leqslant \tau/\gamma \leqslant 9$, the CTPHEUR1 solution deviates from the optimum by $[87(\tau/\gamma) + 398]/[87(\tau/\gamma) + 396] - 1$ with a maximum deviation of 0.23%.

For $9 \leqslant \tau/\gamma \leqslant 9.5$, the CTPHEUR1 solution deviates from the optimum by $[83(\tau/\gamma) + 434]/[87(\tau/\gamma) + 396] - 1$ with a maximum deviation of 0.17%.

References

[1] Booth H, Westbrook J. A linear algorithm for analysis of minimum spanning and shortest-path trees of planar graphs. Algorithmica 1994;11:341–52.

[2] Chen G, Zhang G. A constrained minimum spanning tree problem. Computers & Operations Research 2000;27:867–75.

[3] Cook WJ, Cunningham WH, Pulleyblank WR, Schrijver A. Combinatorial optimization. New York: Wiley, 1998.

[4] Cormen TH, Leiserson CE, Rivest RL. Introduction to algorithms. New York: McGraw-Hill, 1994.

[5] Dijkstra EW. A note on two problems in connection with graphs. Numerische Mathematik 1959;1:269–71.

[6] Eppstein D. Shortest path along an MST. Computational Theory 1999.

[7] Eppstein D. Email communication, June 3, 1999.

[8] Garey MR, Johnson DS. Computers and intractability: a guide to the theory of NP-completeness. San Francisco, CA: W.H. Freeman and Co, 1979.
[9] Gross J, Yellen J. Graph theory and its applications. New York: CRC Press, 1999.
[10] Hillier FS, Lieberman GJ. Introduction to operations research, 6th Edition. New York: McGraw-Hill, 1995.
[11] Khuller S, Raghavachari B, Young N. Balancing minimum spanning trees and shortest-path trees. Algorithmica 1995;14:305–21.
[12] Kruskal Jr. JB. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of ACM 1956;7/1:48–50.
[13] Nemhauser GL, Wolsey LA. Integer and combinatorial optimization. New York: Wiley, 1988.
[14] Papadimitriou CH, Steiglitz K. Combinatorial optimization: algorithms and complexity. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[15] Prim RC. Shortest connection networks and some generalizations. Bell System Technical Journal 1957;36:1389–401.
[16] Rosen KH. Discrete mathematics and its applications, 2nd Edition. New York: McGraw-Hill, 1991.
[17] Saltzman MJ. Counterexample to minimum spanning tree = shortest path tree? Usenet News Group Sci. op-research, 31 May 1995.
[18] Winston WL. Operations research: applications and algorithms, 3rd Edition. Belmont, CA: International Thomson Publishing, 1994.
[19] Winston WL. Introduction to mathematical programming: applications and algorithms, 2nd Edition. Belmont, CA: International Thomson Publishing, 1995.
[20] Wolsey LA. Integer programming. New York: Wiley, 1998.
[21] Zhou G, Gen M. Genetic algorithm approach on multi-criteria minimum spanning tree problem. European Journal of Operational Research 1999;114:141–52.

Francis J. Vasko is Professor of Mathematics and Computer and Information Science at Kutztown University of Pennsylvania. He worked at Bethlehem Steel and continues to serve them as a consultant. His current research focuses on using mixtures of combinatorial optimization techniques to more accurately model and solve important real-world applications in production planning, strategic planning, and resource allocation. He earned a BS in mathematics education from Kutztown University, and an MS in mathematics, an MSIE (Operations Research), and a Ph.D. in Operations Research from Lehigh University. He has over 30 journal publications.

Robert S. Barbieri received his B.S. in mathematics from Kutztown University and is currently pursuing his M.S. in Industrial Engineering at Lehigh University. His primary research interests are in operations research.

Brian Q. Rieksts received his B.S. in mathematics from Kutztown University and is currently pursuing his M.S. in Industrial Engineering at Pennsylvania State University. His research interests are in graph theory and networks, in particular, minimal broadcast networks.

Kenneth L. Reitmeyer received a B.S. degree in computer science from Kutztown University and is currently pursuing his M.S. in computer science from Lehigh University and is employed in the Research Department of Bethlehem Steel Corporation.

Kenneth L. Stott, Jr. was the Supervisor of the Systems Analysis Group in the Research Department of the Bethlehem Steel Corporation. (He retired on 7-31-00). He earned a BSEE from Drexel University, an MSEE from Stevens Institute, an MSIE and a Ph.D (Operations Research) from Lehigh University in 1971. Dr. Stott was an Adjunct Professor at Lehigh University for 16 years.