Received: 28 December 2021 | Revised: 22 January 2022 | Accepted: 28 January 2022 | Published online: 29 January 2022

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2022, Vol. 1(1) 13–20
DOI: 10.47852/bonviewJCCE208918205514

A Simple Methodology that Efficiently Generates All Optimal Spanning Trees for the Cable-Trench Problem

Francis J. Vasko[1,*], Yun Lu[1] and Bryan McNally[1]

[1] *Kutztown University, USA*

Abstract: The Cable-Trench Problem (CTP) is defined as a combination of the Shortest Path and Minimum Spanning Tree Problems. Specifically, let $G = (V,\ E)$ be a connected weighted graph with specified vertex $v_1 \in V$ (referred to as the *root*), length $l(e) \geq 0$ for each $e \in E$, and positive parameters τ and γ. The Cable-Trench Problem is the problem of finding, for given values of τ and γ, a spanning tree T of G 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_1 to all other vertices of V. 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. This is the first article to present a methodology that iteratively uses integer programming software (CPLEX in this article) to efficiently generate *all optimal* spanning trees (GEAOST) for a CTP (for all values of R). An example will illustrate how sensitive the spanning trees solution can be to small changes in edge lengths. Also, GEAOST will be used to generate all optimal spanning trees for graphs based on a real-world radio astronomy application. How the sequence of all optimal spanning trees can be used for sensitivity analysis will be demonstrated.

Keywords: cable-trench problem, integer programming software, CPLEX®, radio astronomy application, minimum spanning tree, shortest path, sensitivity analysis

1. Introduction

Vasko et al. (2002) defined the Cable-Trench Problem (CTP) as a way to combine the shortest path spanning tree problem and the minimum spanning tree problem in a weighted graph with a specified root. The name "Cable-Trench" comes from the fact that a physical application of the CTP is the problem of minimizing the cost to create a campus network in which each building on a campus is connected to a central server with its own dedicated, underground cable. Since the seminal CTP paper (Vasko et al., 2002), there have been a number of publications describing the applications and extensions of the CTP. A representative sample of these publications will be discussed in the literature review section.

Formally, let $V = \{v_1, \ldots,\ v_n\}$ be a set of vertices, and let $E \subseteq \{(v_i, v_j) | 1 \leq i, j \leq n,\ i \neq j\}$ be a set of edges. As defined in Vasko et al. (2002), for a connected weighted graph $G = (V,\ E)$ with specified root vertex $v_1 \in V$, let $l(e) \geq 0$ be the length of the edge $e \in E$, and let τ and γ denote positive weighting parameters. For given values of τ and γ, a solution to the CTP is any spanning tree T of G that minimizes $\tau l_\tau(T) + \gamma l_\gamma(T)$, where $l_\tau(T)$ is the total length of T, and $l_\gamma(T)$ is the total path length in T from v_1 to all other $v_k \in V$. The cost of digging the trench is τ per unit length, and the cost of the cable required is γ per unit length.

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.

In Vasko et al. (2002), a heuristic is developed to solve the CTP for all values of $R = \tau/\gamma$. The main contribution of this current article is that it is the first to present a methodology that iteratively uses integer programming software (CPLEX in this article, but Gurobi and others can be used) to efficiently generate *all optimal* spanning trees for a CTP (for all values of R). Additionally, how this sequence of all optimal spanning trees can be used for sensitivity analysis will be demonstrated. It is important to note that, up to this point, all the applications and extensions of the CTP have dealt with solving a CTP at *particular* values for τ and γ. In other words, applications and extensions of the CTP have focused on particular unit "cable" and unit "trench" costs.

In the next section, publications dealing with the CTP will be discussed. Then some important properties of the CTP will be reviewed. This will be followed by a presentation of our methodology for generating all optimal spanning trees (GEAOST) for a CTP. An example will be used to illustrate this methodology. Next, GEASOST will be used to demonstrate how sensitive the spanning trees in a solution can be to small changes in edge lengths. This will be followed by GEAOST used to solve for all optimal CTP spanning trees for graphs based on an actual radio astronomy application. Next, how the sequence of all optimal spanning trees can be used for sensitivity analysis will be demonstrated. Finally, a summary and potential future work will be provided.

*Corresponding author: Francis J. Vasko, Kutztown University, USA. Email: vasko@kutztown.edu

2. Literature Review

Since the seminal CTP paper (Vasko et al., 2002), there have been a number of publications describing applications of the CTP to model analogous situations. Examples include the application of the CTP to significantly reduce the cost to upgrade and deploy wired and wireless access networks by Nielsen et al. (2008). Marianov et al. (2012) formulated the p-cable trench problem (p-CTP) and generalized the CTP to forests (disjoint unions of trees) to optimize the construction of roads and sawmills for a logging operation and to optimize the construction of canals and wells for irrigation. Both Calik et al. (2017) and Lalla-Ruiz et al. (2016) developed solution procedures for solving the p-CTP based on mathematical programming concepts. Gutiérrez-Jarpa et al. (2015) and Marianov et al. (2012) extended the p-CTP to the p-cable trench problem with covering (p-CTPC) that has an application of locating Wi-Fi antennas in Viña del Mar, Chile. Jamill and Ramezankhani (2015) used the CTP formulation to route power transmission in a metro depot. Schwarze (2015) defines the Multi-Commodity Cable Trench Problem in which a network structure is designed such that different cable types (commodities) are inserted into the same trenches. Most recently, Schwarze et al. (2021) further generalized the problem approach of Marianov et al. (2012) and presented the capacitated cable trench problem with facility costs (cCTP-FL), where the opened facilities have a cost and limitation on the number of associated customers, so that facility location decisions in wire-based networks can be taken under a more realistic cost scenario. The CTP is generalized and used to solve a nontrivial application to vascular image analysis by Jiang et al. (2011) and Vasko et al. (2015).

An interesting logistical problem formulated as a CTP by Girard (2013) and Zyma et al. (2017) is the problem of connecting 96 low-frequency antenna arrays forming a new radio telescope distributed across a 400×450-meter area in the Nançay radio observatory in France to a central control facility via coaxial cables. To protect the cables, trenches are dug for the cables to run underground. Also, any number of cables can be laid in a given trench. A minimum cost (combining both cable and trench costs) configuration will necessarily be a spanning tree of the 96 antenna arrays with the central control facility as the root. For brevity, we refer to this problem as the radio astronomy antenna connection problem (RAACP). In a subsequent section, we will define several graphs based on the RAACP and solve for all optimal spanning trees using the GEAOST algorithm.

All of the applications and extensions of the CTP mentioned above have dealt with solving a CTP at particular values for τ and γ. In other words, applications and extensions of the CTP have focused on particular unit cable and unit trench costs. Researchers (Jeng et al., 2006, 2007) have attempted to use DNA-based evolutionary computing to solve small CTPs. The largest CTP-type problems solved (approximately) involved the vascular imaging application (Vasko et al., 2015). Stochastic greedy and semi-greedy based heuristics were used to efficiently solve, for particular values of τ and γ, graphs with up to 25,000 vertices and 11 million edges. For these instances, 60 solutions were generated in about 30 min on a standard PC, and the best solution among the 60 was chosen as the answer. In terms of applications using exact solution methods, the radio astronomy application (Zyma et al., 2017) used CPLEX to solve, again for particular values of τ and γ, graphs with 97 vertices (root node and 96 antenna arrays), and almost 3000 edges in about 10 min on a standard PC.

Because we will be using the sequence of all optimal spanning trees to perform sensitivity analyses on the ratio of trench to cable cost, we will now provide a brief background on linear programming and mixed integer programming sensitivity analysis. Classic linear programming sensitivity analysis (Hillier & Lieberman, 2010; Winston, 2004) deals with determining how much an objective function coefficient or constraint right-hand side coefficient can be perturbed, and the current optimum remains the same. This analysis assumes that only one coefficient is modified at a time. The 100% Rule (Bradley et al., 1977) for linear programming analyzes the impact of several objective function coefficients or several constraint right-hand sides changing at the same time. Sensitivity analysis results for mixed integer linear programming problems (MILP), such as the MFCTP, are more limited and largely derived from the idea of inference duality (Cook et al., 1986; Dawande & Hooker, 2000). The inference dual of a MILP asks how the optimal value can be deduced from the constraints. Yi and Lu (2019) studied MILP sensitivity analysis applied to haul road layout design for earth-moving operations.

3. Important Solution Properties of the Cable-Trench Problem

In this section, we will summarize some important properties (first discussed in Vasko et al. (2002)) of the spanning trees in the CTP solution over all values of $R = \tau/\gamma$ as R varies from 0 to ∞. Let T1 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 $l_\tau(\mathrm{T1})$ be the total trench length corresponding to T1, and let $l_\gamma(\mathrm{T1})$ be the total cable length corresponding to T1. Let TΩ be a minimum spanning tree that minimizes the total path length from v_0 to all other vertices in V, and let $l_\tau(\mathrm{T}\Omega)$ be the total trench length corresponding to TΩ, and let $l_\gamma(T\Omega)$ be the total cable length corresponding to $T\Omega$. If $l_\tau(\mathrm{T1}) = l_\tau(T\Omega)$, then T1 is the optimal spanning tree for all values of $\tau/\gamma > 0$. Otherwise, $l_\tau(\mathrm{T1}) > l_\tau(T\Omega)$, $l_\gamma(\mathrm{T1}) < l_\gamma(T\Omega)$, and $\mathrm{T1} \neq T\Omega$.

If $\mathrm{T1} \neq T\Omega$, then the optimal solution to the CTP is a sequence of spanning trees such that as the τ/γ value increases, the total length of the spanning trees *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.

4. GEAOST Algorithm and Illustrated Example

4.1. Introduction

The heuristic outlined in Vasko et al. (2002) starts by generating the spanning tree T1 and then uses a neighborhood search methodology to heuristically generate a sequence of spanning trees. Although this heuristic performed well on the examples analyzed in Vasko et al. (2002), it typically did not generate all the optimal spanning trees.

Given the current power of integer programming software and PC hardware, the main purpose of this article is to demonstrate a methodology we refer to as GEAOST that will efficiently generate all optimal spanning trees for graphs with up to 30 nodes and 68 edges in at most 6 min on a standard PC for all values of $R = \tau/\gamma$ in $[0,\infty)$. To solve a CTP for specified values of τ and γ, the mathematical formulation for the CTP (MFCTP) from Section 3.1 in Zyma et al. (2017) will be used.

Without loss of generality, we assume that $\gamma = 1$, and we will determine all optimal spanning trees for τ in $[0, \infty)$. T1 and TΩ are as defined previously. Now define $\mathrm{CF}(\mathrm{Ti}(\tau))$ to be the total CTP cost function for spanning tree Ti at τ and $\mathrm{CF}(\mathrm{T}\tau_i(\tau))$ to be the total CTP cost function at τ for the spanning tree that is optimum (minimum cost) when $\tau = \tau_i$. We will now outline the GEAOST logic. Detailed pseudo-code and a flowchart (Figure A1) are provided in an appendix.

4.2. Outline of GEAOST logic

STEP1: Determine T1 and TΩ using MFCTP discussed previously.

STEP2: If T1 = TΩ, then T1 is the optimal spanning tree for all τ in $[0, \infty)$ and terminate.

If T1 ≠ TΩ, label T1 as LSST (left-side spanning tree) and TΩ as RSSP (right-side spanning tree) and go to STEP3.

STEP3: Let τ_0 be the breakeven point for LSSP and RSSP. That is, $CF(LSSP(\tau_0)) = CF(RSSP(\tau_0))$.

Solve MFCTP at τ_0 (recall $\gamma = 1$). If the optimal cost is the same as $CF(LSSP(\tau_0))$, then γ_0 is an optimal breakeven point, and no spanning tree needs to be added to the solution.

However, if the optimal spanning tree has a smaller cost than LSSP = RSSP, then this spanning tree denoted as BETTER must be inserted between LSSP and RSSP.

STEP4: Now determine the breakeven points for LSSP and BETTER and for BETTER and RSSP, and insert into the solution the optimal spanning trees at the breakeven points if the costs are improved by these spanning trees. Otherwise, these points are optimal breakeven points.

STEP5: Continue this process until all breakeven points are optimum. The resulting spanning trees and breakeven points will be an optimal CTP solution over all values of τ in $[0, \infty)$.

It is obvious that by inserting "enough" spanning trees, for each τ in $[0, \infty)$, there will be a spanning tree in the GEAOST ordered set of spanning trees denoted as $T\tau$ such that the difference between the optimal MFCTP objective function value for τ and the objective function value of $T\tau$ at τ can be made arbitrarily small.

GEAOST was programmed using the control language JavaScript, and the MIP language was OPL (Optimization Programming Language). All executions of CPLEX were on a standard PC with the following specifications: 16 GB of memory, Intel processor with 2.9 GHz, 1000 GB hard drive, and Windows 10.

4.3. Example solved using GEAOST

The following problem, which is Example 2 in Vasko et al. (2002), will illustrate the logic of GEAOST.

Specifically, let $V = \{1,2,3,4,5,6,7\}$ 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.

GEAOST Solution: As previously, assume $\gamma = 1$.

$CF(T1(\tau)) = 449 + 279\tau$ and $CF(T\Omega(\tau)) = 610 + 180\tau$.

Setting $CF(T1(\tau)) = CF(T\Omega(\tau))$ implies that $\tau = 161/99$.

Solving the MFCTP (given above) for $\tau = 161/99$ and $\gamma = 1$ gives a new spanning tree with $CF(T(161/99)(\tau)) = 469 + 219\tau$.

Since $CF(T(161/99)(161/99)) < CF(T1(161/99)) = CF(T\Omega(161/99))$, the spanning tree T(161/99) needs to be inserted in the solution between T1 and TΩ.

Now we need to determine the breakeven points between T1 and T(161/99) and between T(161/99) and TΩ.

First find the breakeven points between T1 and T(161/99).

$CF(T1(\tau)) = CF(T(161/99)(\tau))$ implies $\tau = 1/3$ and $CF(T(1/3)(\tau) = 4670 + 230\tau$.

Since $CF(T(1/3)(1/3)) < CF(T1(1/3)) = CF(T(161/99)(1/3))$, the spanning tree T(1/3) needs to be inserted in the solution between T1 and T(161/99).

First, we find the breakeven point between T(161/99) and TΩ.

$CF(T(161/99)(\tau)) = CF(T\Omega(\tau))$ implies $\tau = 141/39$.

Since $CF(T(161/99)(141/39)) = CF(T\Omega(141/39))$, 141/39 is an optimal breakpoint between T(161/99) and TΩ.

Now we insert T(1/3) between T1 and T(161/99).

Start by finding the breakeven point between T1 and T(1/3).

$CF(T1\ (\tau)) = CF(T(1/3)(\tau)$ implies $\tau = 11/49$.

Since $CF(T(11/49)(\tau)) = CF(T(1/3)(\tau))$, 11/49 is an optimal breakpoint between T1 and T(1/3).

Now determine the breakeven point between T(1/3) and T(161/99).

$CF(T(1/3(\tau)) = CF(T(161/99)(\tau)$ implies $\tau = 9/11$.

Since $CF(T(9/11)(\tau)) = CF(T(161/99)(\tau))$, 9/11 is an optimal breakpoint between T(1/3) and T(161/99).

All optimal breakpoints have been determined and the optimal spanning trees are as follows:

T1 with $CF(T1(\tau)) = 449 + 279\tau$ for $0 \leq \tau \leq 11/49$,

T(11/49) with $CF(T(11/49)(\tau)) = 460 + 230\tau$ $11/49 \leq \tau \leq 9/11$,

T(9/11) with $CF(T(9/11)(\tau)) = 469 + 219\tau$ $9/11 \leq \tau \leq 141/39$,

T(141/39) = TΩ with $CF(T\Omega\ (\tau)) = 610 + 180$ $141/39 \leq \tau$.

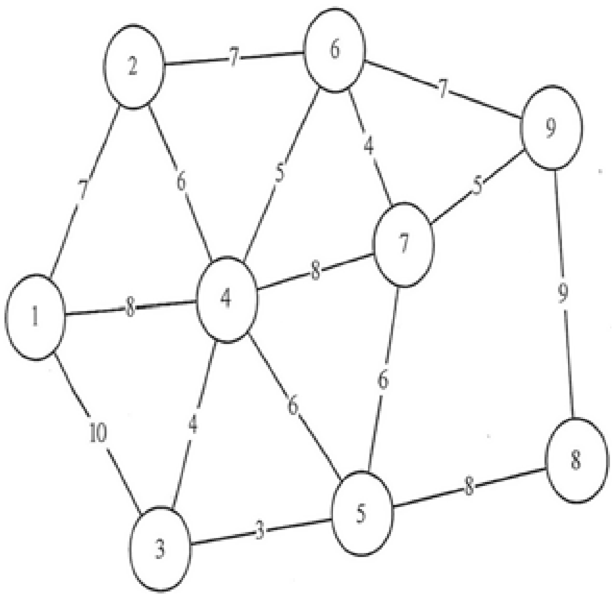
In this example, GEAOST solved the MFCTP integer programming formulation five times (five breakeven points) and determined three optimal break points. For this example, the heuristic in Vasko et al. (2002) did not find optimal spanning trees for τ in the interval (9/11, 141/39).

We will next use Example 4 from Vasko et al. (2002) to demonstrate how sensitive the optimal CTP solution of spanning trees can be to small changes in edge lengths.

4.4. An example of CTP solution sensitivity to edge length changes

We will now use Example 4 from Vasko et al. (2002) to demonstrate how the optimal CTP spanning trees solution can be significantly impacted by just changing the edge length of one edge one unit in length. Example 4 is the following problem. For the graph $G = (V, E)$, 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 graph is given below (See Figure 1).

Figure 1
Graph of Example 4 (Vasko et al., 2002)



The GEAOST CTP spanning trees solution and the intervals over which each spanning tree is optimum are given in Table 1.

Table 1
CTP optimal spanning trees for Example 4 (Vasko et al., 2002)

Spanning trees	Optimal intervals
{(1.2), (1,3), (1,4), (3,5), (4,6), (4,7), (5.8), (6,9)}	$0 \le \tau \le 1/4$
{(1.2), (1,3), (1,4), (3,5), (4,6), (5,8), (6.7), (6,9)}	$1/4 \le \tau \le 1$
{(1.2), (1,4), (3,4), (3,5), (4,6), (5,8), (6,7), (7,9)}	$1 \le \tau \le 7$
{(1.4), (2,4), (3,4), (3,5), (4,6), (5,8), (6,7), (7,9)}	$7 \le \tau \le 28$
{(1.2), (2,4), (3,4), (3,5), (4,6), (5,8), (6,7), (7,9)}	$28 \le \tau$

Now suppose the length of edge (3,5) is reduced just one unit from 3 to 2. In this case, the GEAOST CTP spanning trees solution and the intervals over which each spanning tree is optimum remain *exactly the same* as for the original Example 4. However, suppose instead of reducing the length of edge (3,5) one unit, the length is increased one unit from 3 to 4. In this case, the GEAOST CTP spanning trees solution now contains *seven* spanning trees instead of five. The GEAOST CTP spanning trees solution and the intervals over which each spanning tree is optimum when the length of edge (3,5) is increased to four units are given in Table 2 with the two new spanning trees denoted in bold.

Table 2
CTP optimal spanning trees for Example 4 (Vasko et al., 2002) with (3,5) edge length equal to 4

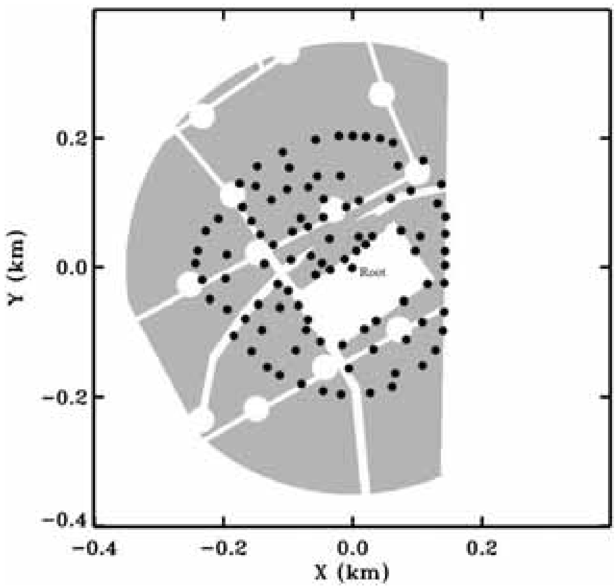
Spanning trees	Optimal intervals
{(1.2), (1,3), (1,4), (3,5), (4,6), (4,7), (5.8), (6,9)}	$0 \le \tau \le 1/4$
{(1.2), (1,3), (1,4), (3,5), (4,6), (5,8), (6.7), (6,9)}	$1/4 \le \tau \le 1/2$
{(1.2), (1,4), (3,4), (4,), (4,6), (5,8), (6.7), (6,9)}	$1/2 \le \tau \le 1$
{(1.2), (1,4), (3,4), (4,5), (4,6), (5,8), (6.7), (7,9)}	$1 \le \tau \le 2$
{(1.2), (1,4), (3,4), (3,5), (4,6), (5,8), (6,7), (7,9)}	$2 \le \tau \le 7$
{(1.4), (2,4), (3,4), (3,5), (4,6), (5,8), (6,7), (7,9)}	$7 \le \tau \le 28$
{(1.2), (2,4), (3,4), (3,5), (4,6), (5,8), (6,7), (7,9)}	$28 \le \tau$

Hence, this small edge length change of one unit for just one edge resulted in two additional spanning trees being added to the optimal CTP spanning trees solution. The GEAOST solutions for Examples 2 and 4 required just a few seconds on the PC specified above and used CPLEX to solve the MFCTPs.

5. Examples Based on a Radio Astronomy Application

As briefly mentioned earlier, an interesting logistical problem formulated as a CTP by Girard (2013) and solved for particular values of τ and γ by Zyma et al. (2017) is the problem of connecting 96 low-frequency (LF) antenna arrays forming a new radio telescope distributed across a 400×450-meter area in the Nançay radio observatory in France to a central control facility via coaxial cables. To protect the cables, trenches will be dug for the cables to run underground. Also, any number of cables can be laid in a given trench. In order to preserve the nature and quality of the signal (i.e., no analog to digital conversion, nor multiplexing), each of the 96 antenna arrays must be connected directly to the central control facility, which will digitize the signals. A minimum cost (combining both cable and trench costs) configuration will necessarily be a spanning tree of the 96 antenna arrays with the central control facility as the root. For brevity, we refer to this problem as the radio astronomy antenna connection problem (RAACP). The layout of the antennas and central processing station (labeled "Root") is given in Figure 2.

Figure 2
Layout of 96 LF antennas, with the central processing station (Girard 2013)



Each black dot in Figure 2 represents a low-frequency antenna array composed of 19 LF antennas arranged in a hexagon that needs to be connected to the central station.

In this article, we will define two CTPs based on actual data from this radio astronomy application. Specifically,

Radio Astronomy Problem 1 (RSP1): Is the CTP formed using the root node and the nine closest antenna arrays (10 nodes)? There are 22 edges each of length less than 200 meters.

Radio Astronomy Problem 2 (RSP1): Is the CTP formed using the root node and the 29 closest antenna arrays (30 nodes)? There are 68 edges each of length less than 100 meters.

The GEAOST solutions for these two problems are given in Tables 3 and 4, respectively.

Table 3
CTP optimal spanning trees cost functions for RSP1

Spanning trees cost functions ($\gamma = 1$)	Optimal intervals
$1047\tau + 1412$	$0 \leq \tau \leq 0.290322581$
$954\tau + 1439$	$0.290322581 \leq \tau \leq 0.297297297$
$917\tau + 1450$	$0.297297297 \leq \tau \leq 2.085106383$
$870\tau + 1548$	$2.085106383 \leq \tau \leq 2.625$
$838\tau + 1632$	$2.625 \leq \tau \leq 6.35$
$818\tau + 1759$	$6.35 \leq \tau \leq 15.2$
$808\tau + 1911$	$15.2 \leq \tau$

Table 4
CTP optimal spanning trees cost functions for RSP2

Spanning trees cost functions ($\gamma = 1$)	Optimal intervals
$2024\tau + 5662$	$0 \leq \tau \leq 0.09375$
$1992\tau + 5665$	$0.09375 \leq \tau \leq 0.25$
$1984\tau + 5667$	$0.25 \leq \tau \leq 0.260869565$
$1961\tau + 5673$	$0.260869565 \leq \tau \leq 0.428571429$
$1919\tau + 5691$	$0.428571429 \leq \tau \leq 0.5$
$1865\tau + 5718$	$0.5 \leq \tau \leq 0.52$
$1840\tau + 5731$	$0.52 \leq \tau \leq 0.625$
$1824\tau + 5741$	$0.625 \leq \tau \leq 0.8125$
$1776\tau + 5780$	$0.8125 \leq \tau \leq 0.851851852$
$1722\tau + 5826$	$0.851851852 \leq \tau \leq 1.13333$
$1707\tau + 5843$	$1.13333 \leq \tau \leq 1.25$
$1691\tau + 5863$	$1.25 \leq \tau \leq 1.782608696$
$1668\tau + 5904$	$1.782608696 \leq \tau \leq 2.551724138$
$1610\tau + 6052$	$2.551724138 \leq \tau \leq 6.53125$
$1578\tau + 6261$	$6.53125 \leq \tau \leq 9.545454545$
$1567\tau + 6366$	$9.545454545 \leq \tau \leq 12.9090909$
$1545\tau + 6650$	$12.9090909 \leq \tau \leq 16.166667$
$1533\tau + 6844$	$16.166667 \leq \tau \leq 35.2$
$1528\tau + 7020$	$35.2 \leq \tau \leq 88.3125$
$1512\tau + 8433$	$88.3125 \leq \tau$

As indicated in Table 3, GEAOST generated seven optimal spanning trees for RSP1. This required about a minute of execution time. In Table 4, summary information for the 20 optimal spanning trees generated by GEAOST for RSP2 is provided. The execution time for GEAOST to completely solve RSP2 was about 6 min.

6. CTP Sensitivity Analysis

Although efficiently generating all optimal spanning trees to a CTP may seem more theoretical than practical, we will now demonstrate how these optimal spanning trees can be used to perform straightforward practical sensitivity analyses dealing with perturbations in the per unit cable and trench costs of a CTP. Suppose that for a certain application, there was a need to solve the CTP mixed integer programming problem (MFCTP) at particular values for τ and γ. To illustrate, let us consider three τ/γ ratios of 2, 5, and 8. In other words, suppose that three scenarios are being considered in which the per unit trench cost was 2, 5, and 8 times the per unit cable cost. The MFCTP solved at each of these ratios (assume $\gamma = 1$) will give *one* optimal spanning tree for each ratio, but how sensitive are these solutions to these specific ratio values? In Table 5, information from the GEAOST optimal spanning trees solutions for Example 2, Example 4, Example 4 modified, RSP1, and RSP2 is given demonstrating how interval sensitive these solutions are at the ratios 2, 5, and 8.

The entries in Table 5 give the intervals for τ over which the optimal spanning tree found by solving the MFCTP remains the optimum. For instance, in Example 4, the optimal spanning tree remains optimum as long as τ is in the interval [1, 7] or in the interval [7, 28]. Hence, a ratio of 2 or 5 can vary and still have the same optimal spanning tree as long as the ratio value is in the interval [1, 7]. Furthermore, for Example 4, the optimal spanning tree for the ratio value of 8 remains optimum for τ in the interval [7, 28]. So, if τ is expected to have a value of 8 but is actually less than 7, the optimal spanning tree at a ratio of 8 is no longer optimum. Let us now consider the modified Example 4 where edge (3, 5) has length 4 instead of 3. In this case, there are two optimal spanning trees when the ratio is exactly 2. However, if the ratio is even slightly less than 2 or slightly greater than 2, then there is only one optimal spanning tree, and it differs based on if the ratio is less than or greater than 2. Finally, consider the first CTP (RSP1) based on the radio astronomy application. In this case, for the ratio of 2, the optimal spanning tree remains optimum if τ is in the interval [0.2973, 2.0851]. However, once τ is greater than 2.0851, the optimal spanning tree changes and remains optimal as long as τ is in the interval [2.0851, 2.625].

7. Summary and Future Work

The CTP, first defined by Vasko et al. (2002) in 2002, has been shown in numerous publications (as illustrated in the literature review section) to have diverse real-world applications beyond the original application of digging trenches and laying cables in a minimum cost manner. However, these applications typically

Table 5
CTP sensitivity analyses

	Ratios (assumes $\gamma = 1$)		
CTPs	2	5	8
Example 2	$9/11 \leq \tau \leq 141/39$	$\tau \geq 141/39$	$\tau \geq 141/39$
Example 4	$1 \leq \tau \leq 7$	$1 \leq \tau \leq 7$	$7 \leq \tau \leq 28$
Example 4 (modified)	$1 \leq \tau \leq 2$	$2 \leq \tau \leq 7$	$7 \leq \tau \leq 28$
	$2 \leq \tau \leq 7$		
RSP1	$0.2973 \leq \tau \leq 2.0851$	$2.625 \leq \tau \leq 6.35$	$6.35 \leq \tau \leq 15.2$
	$2.0851 \leq \tau \leq 2.625$		
RSP2	$1.7826 \leq \tau \leq 2.5517$	$2.5517 \leq \tau \leq 6.5312$	$6.5313 \leq \tau \leq 9.5454$

solve a problem for particular "cable" and "trench" per unit costs. This is the first article to document an efficient algorithm (GEAOST) that generates *all guaranteed optimal* spanning trees for a CTP. Furthermore, it was demonstrated how the sequence of all optimal spanning trees can be used for sensitivity analysis.

Applications modeled as CTPs with graphs having up to 30 nodes and 68 edges were successfully solved by the GEAOST algorithm generating all optimal spanning trees in at most 6 min on a standard PC. Sensitivity analyses performed on five CTPs discussed in this article confirmed the practical benefit of generating all optimal spanning trees for a CTP. We intend to refine GEAOST and use it to completely solve larger CTPs. Also, we plan on using different integer programming software. Specifically, we plan to use the latest version of Gurobi (9.5) as the "engine" for GEAOST.

Acknowledgments

The authors would like to acknowledge Dr. Kenneth L. Stott, Jr. for his helpful discussions with the authors concerning the content of this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

References

Bradley, S., Hax, A., & Magnanti, T. (1977). *Applied mathematical programming*. USA: Addison-Wesley.

Calik, H., Leitner, M., & Luipersbeck, M. (2017). A Benders decomposition based framework for solving cable trench problems. *Computers and Operations Research*, *81*, 128–140. https://doi.org/10.1016/j.cor.2016.12.015

Cook, W., Gerards, A. M. H., Schrijver, A., & Tardos, E. (1986). Sensitivity theorems in integer programming. *Mathematical Programming*, *34*, 251–264. https://doi.org/10.1007/BF01582230

Dawande, M. W., & Hooker, J. N. (2000). Inference-based sensitivity analysis for mixed integer/linear programming. *Operations Research*, *48*(4), 623–634. https://doi.org/10.1287/opre.48.4.623.12420

Girard, J. N. (2013). *Développement de la Super Station LOFAR et observations planétaires avec LOFAR*, PhD thesis, Observatoire de Paris. Retrieved from: https://tel.archives-ouvertes.fr/tel-00835834.

Gutiérrez-Jarpa, G., Obreque, C., Marianov, V., & Contreras, A. (2015). P-cable trench problem with covering. *SSRN*. https://doi.org/10.2139/ssrn.2814676

Hillier, F. S., & Lieberman, G. J. (2010). *Introduction to operations research*, USA: McGraw-Hill.

Jamill, A., & Ramezankhani, F. (2015). An extended mathematical programming model to optimize the cable trench route of power transmission in a metro depot. *International Journal of Transportation Engineering*, *3*(2), 109–123. https://doi.org/10.22119/ijte.2015.13837

Jeng, D. J. F., Kim, I., & Watada, J. (2006). DNA-based evolutionary algorithm for cable trench problem. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 922–929. https://doi.org/10.1007/11893011_117

Jeng, D.F., Kim, I., & Watanda, J. (2007). Bio-inspired evolutionary method for cable trench problem. *International Journal of Innovative Computing, Information and Control*, *3*(1), 111–118. https://doi.org/10.1.1.104.1700

Jiang, Y., Zhuang, Z. W., Sinusas, A. J., Staib, L. H., & Papademetris, X. (2011). Vessel connectivity using Murray's hypothesis. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011: 14th International Conference, Toronto, Canada, September 18-22, 2011, Proceedings, Part III 14*, 528–536. https://doi.org/10.1007/978-3-642-23626-6_65

Lalla-Ruiz, E., Schwarze, S., & Voss, S. (2016). A matheuristic approach for the p-cable trench problem. In P. Festa., M. Sellmann & J. Vanschoren (Eds.), *Learning and intelligent optimization* (pp. 247–252). Springer. https://doi.org/10.1007/978-3-319-50349-3_19

Marianov, V., Gutiérrez-Jarpa, G., Obreque, C., & Cornejo, O. (2012). Lagrangean relaxation metaheuristics for the *p*-cable-trench problem. *Computers and Operations Research*, *39*(3), 620–628. https://doi.org/10.1016/j.cor.2011.05.015

Nielsen, R. H., Riaz, M. T., Pedersen, J. M., & Madsen, O. B. (2008). On the potential of using the cable trench problem in planning of ICT access networks. In *2008 50th International Symposium ELMAR*, *2*, 585–588.

Schwarze, S. (2015). The multi-commodity cable trench problem. In *European Conference on Information Systems*, Münster, Germany. https://doi.org/10.18151/7217472

Schwarze, S., Lalla-Ruiz, E. & Voß, S. (2021). Modeling the capacitated *p*-cable trench problem with facility costs. *Central European Journal of Operations Research*, *29*, 713–735. https://doi.org/10.1007/s10100-020-00674-w

Vasko, F. J., Barbieri, R. S., Rieksts, B. Q., Reitmeyer, K. L., & Stott, Jr. K. L. (2002). The cable trench problem: Combining the shortest path and minimum spanning tree problems. *Computers & Operations Research*, *29*(5), 441–458. https://doi.org/10.1016/S0305-0548(00)00083-6

Vasko, F., Landquist, E., Kresge, G., Tal, A., Jiang, Y., & Papademetris, X. (2015). A simple and efficient strategy for solving very large-scale generalized cable-trench problems. *Networks*, *67*(3), 199–208. https://doi. org/10.1002/net.21614

Winston, W.L. (2004). *Operations research: Applications and algorithms*. USA: Thomson Brooks.

Yi, C., & Lu, M. (2019). Mixed-integer linear programming-based sensitivity analysis in optimization of temporary haul road layout design for earthmoving operations. *Journal of Computing in Civil Engineering*, *33*(3), 1–14. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000838

Zyma, K., Girard, J. N., Landquist, E., Schaper, G., & Vasko, F. J. (2017). Formulating and solving a radio astronomy antenna connection problem as a generalized cable-trench problem: An empirical study. *International Transactions in Operational Research*, *24*(5), 943–957. https://doi.org/10.1111/itor.12312

How to Cite: Vasko, F. J., Lu, Y., & McNally, B. (2022). A Simple Methodology that Efficiently Generates All Optimal Spanning Trees for the Cable-Trench Problem. *Journal of Computational and Cognitive Engineering* 1(1), 13–20, https://doi.org/10.47852/bonviewJCCE208918205514

Appendix

GEAOST pseudocode

1. Find (SP/MST) : the strategy first solves the MFCTP with $\tau = 0$, and $\gamma = 1$. The objective function is the minimum cable length. In the MFCTP, now add a constraint that forces the cable length equal to the minimum cable length determined in the previous MFCTP where $\tau = 0$ and $\gamma = 1$, but now set $\tau = 1$ and $\gamma = 0$.
2. Find (MST/SP) : The strategy first solves the MFCTP with $\tau = 1$, and $\gamma = 0$. The objective function is the minimum trench length. In the MFCTP, now add a constraint that forces the trench length equal to the minimum trench length determined in the previous MFCTP where $\tau = 1$ and $\gamma = 0$, but now set $\tau = 0$ and $\gamma = 1$.
3. Label 4 variables capturing the four figures found in steps 1 and 2. They will be referred to as:
   - C_SP/MST where $\tau = 0$ and $\gamma = 1$
   - T_SP/MST where $\tau = 1$ and $\gamma = 0$ with cable length constraint
   - T_MST/SP where $\tau = 1$ and $\gamma = 0$
   - C_MST/SP where $\tau = 0$ and $\gamma = 1$ with trench length constraint
4. Declare the following variables with initial values:
   - count = 1
   - t = 0
   - idx = 2
   - rows = (depends on size of CTP)
   - cols = 4
   - $\tau = 0$
   - $\gamma = 1$
5. Construct empty matrix of size rows X cols we will call M[rows][cols]

GEAOST functions

1. calcT
   returns (M[1][1] − M[0][1])/(M[0][0] − M[1][0])
2. sortFunction
   sorts M on first column descending order
3. overwriteRow
   removes the first row by shifting all rows up
4. displayFull Matrix
   prints out M to log (can toggle off for more compact log report)

GEAOST execution

1. Write the following to the first row in empty matrix M:
   - M[0][0] = T_SP/MST
   - M[0][1] = C_SP/MST
   - M[0][2] = t
   - M[0][3] = (M[0][0] * t) + M[0][1]
2. Print out to log:
   - T_SP/MST(T) + C_SP/MST(C) = matrix[0][3]
   - $\tau/\gamma = t$
3. Write the following to the second row of matrix M:
   - M[1][0] = T_MST/SP
   - M[1][1] = C_MST/SP
   - set t = calcT() then assign M[1][2] = t
   - M[1][3] = (M[0][0] * t) + M[0][1]
4. Print out to log:
   - T_MST/SP(T) + C_MST/SP(C) = matrix[1][3]
   - $\tau/\gamma = t$
5. Main Loop
   WHILE M[0, 0] does not equal T_MST/SP:
   Initiate MIP program and declare the following variables that will be passed back and forth from the MIP program and process control program.
   - $\tau = t$ (is sent to the MIP program)
   - coefT = CoefT (this coefficient for T is retrieved from the MIP program)
   - coefC = CoefC (this coefficient for C is retrieved from the MIP program)
   - OBJ_VAL = objective value solution (this is the solution from MIP program)

IF coefT equals M[0, 0]:
    Print: OPTIMUM FOUND – left side equality
    Print: coefT(T) + coefC(C) = OBJ_VAL
    Print: $\tau/\gamma = t$
    call overwriteRow()
    decrement idx by 1

ELSE IF coefT equals M[1][0]:
    Print: OPTIMUM FOUND – right side equality
    Print: M[0, 0] (T) + M[1][0](C) = OBJ_VAL
    Print: $\tau/\gamma = t$
    call overwriteRow()
    decrement idx by 1

ELSE IF coefT is greater than M[0, 0]:
    Print: coefT is greater than left hand side of equation
    Print: coefT(T) + coefC(C) = OBJ_VAL
    Print: $\tau/\gamma = t$
    call overwriteRow()
    decrement idx by 1

ELSE IF coefT is less than M[1][0]:
    Print: coefT is less than right hand side of equation
    Print: coefT(T) + coefC(C) = OBJ_VAL
    Print: $\tau/\gamma = t$
    call overwriteRow()
    decrement idx by 1
    Update Matrix M
        M[0, 0] = coefT
        M[1][0] = coefC
        M[1][0] = t
        M[1][0] = OBJ_VAL

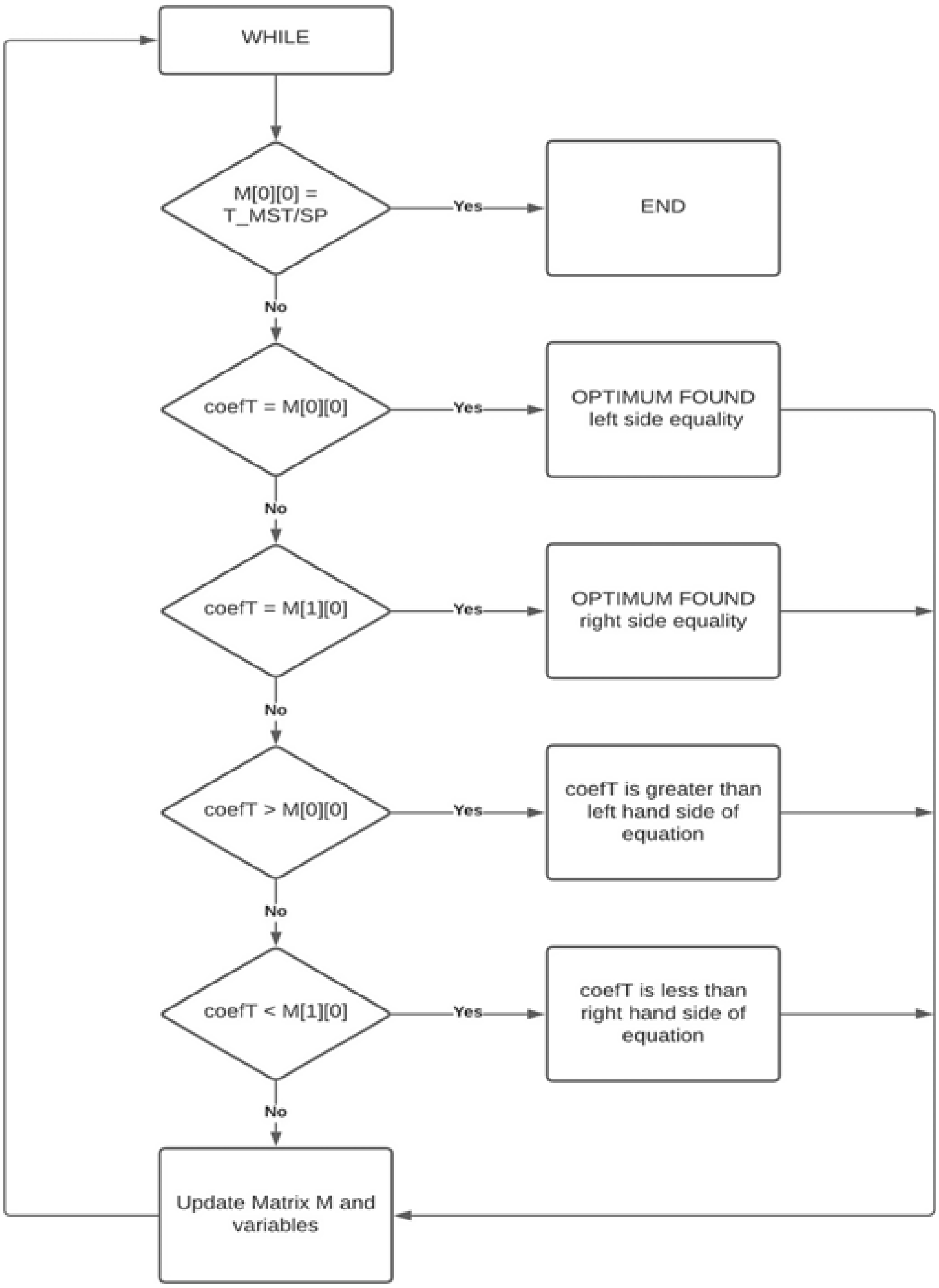
ELSE
    Update Matrix M
        M[idx][0] = coefT
        M[idx[1] = coefC
        M[idx] [1] = t
        M[idx] [1] = OBJ_VAL
    Print: coefT(T) + coefC(C) = OBJ_VAL
    Print: $\tau/\gamma = t$
    increment idx by 1
    call sortFunction()

IF M[0, 0] does equal T_MST/SP:

Print: “END”
Terminate MIP
Break While Loop
END

t = calcT()
increment count by 1
Terminate MIP
Continue While Loop

Figure A1
GEAOST flowchart.



GEAOST Flow chart