U. S. NAVAL RESEARCH LABORATORY

Washington, DC 20375-5320

NRL/FR/5521--20-10,401

Network Sum Power Interference at an External Node

CLEMENT K. KAM
SASTRY KOMPELLA
CALEB Z. BOWERS
JOSEPH P. MACKER

Networks and Communication Systems Branch
Information Technology Division

November 6, 2020

Report Documentation Page	*Form Approved OMB No. 0704-0188*

1. REPORT DATE (DD-MM-YYYY)	**2. REPORT TYPE**	**3. DATES COVERED (From – To)**
06-11-2020	NRL Formal Report	Sep 2019 - Sep 2020

4. TITLE AND SUBTITLE
Network Sum Power Interference at an External Node

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61153N

6. AUTHOR(S)
Clement K. Kam
Sastry Kompella
Caleb Z. Bowers
Joseph P. Macker

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
1P81

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL/FR/5521--20-10,401

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
One Liberty Center
875 N. Randolph Street, Suite 1425
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)
ONR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
DISTRIBUTION A: Approved for public release, distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The purpose of this work is to connect nodes to form a connected ad hoc network, while trying to limit the interference to a node or device that is external to the network caused by the emitted power required to make such network connections. Specifically, we consider a case where the interference at the external node cannot exceed some power threshold, and the goal is to connect as many nodes to form a tree without exceeding the threshold. In this formulation, the cost to connect a pair of nodes is the sum of the interference caused by transmitting in each direction. We call this the Largest Constrained Tree (LCT) problem. We first prove that this problem is NP-complete, and then we formulate it as a mixed integer linear program to be solved computationally. We consider two different cost metrics, for the cases where there is information about the external node(s) location and when there is not. We generate Monte Carlo simulation results for various network sizes, node densities, external node distances, and network activity levels.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSION**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Clement K. Kam
U/U	U/U	U/U	SAR	19	**19b. TELEPHONE NUMBER** (Include area code) 202-767-0047

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18
LaTeX

This page intentionally left blank

CONTENTS

EXECUTIVE SUMMARY E-1

1. BACKGROUND 1
   1.1 Introduction 1
   1.2 Related Work 1
   1.3 Prior Related NRL Work 2

2. SYSTEM MODEL 2
   2.1 Network Model 2

3. LARGEST CONSTRAINED TREE PROBLEM 3
   3.1 Complexity 3
   3.2 Optimization 5

4. SIMULATION RESULTS 6
   4.1 Distance-Based Cost 7
   4.2 External Node-Based Cost 9

5. CONCLUSION AND FUTURE WORK 13

REFERENCES 13

FIGURES

1 Weighted node 4

2 Reduction from Vertex Cover to Largest Weight-Constrained Tree 4

3 8 node example, distance-based cost, $\alpha = 1$ 7

4 Increasing network size, fixed node density, distance-based cost, $\alpha = 1$ 8

5 Increasing network size, increasing node density, distance-based cost, $\alpha = 1$ 8

6 Increasing network activity, distance-based cost 9

7 8 node LCT example, $d = 1$, $\alpha = 1$ 9

8 64 node grid LCT example, $\alpha = 1$ 10

9 Increasing network size, external node-based cost, fixed node density, $\alpha = 1$ 11

10 Increasing network size, external node-based cost, increasing node density, $\alpha = 1$ 11

11 Increasing network activity, distance-based cost 12

12 Increasing external node distance, external node-based cost, $\alpha = 1$ 13

EXECUTIVE SUMMARY

The purpose of this report is to study connecting nodes to form an ad hoc network subject to a limit on the allowable interference power incident on nodes external to the network. Unlike most existing work on limiting interference to other networks, we focus on the physical interference model that accounts for the cumulative effect of multiple nodes in the network transmitting simultaneously. Assuming the network geometry and topology is known, the goal is to connect as many nodes as possible under a sum power interference constraint.

We first formulate the problem of constructing the largest tree subject to a constraint on the sum power experienced by external nodes, which we call the Largest Constrained Tree (LCT) problem. The network is modeled as a graph, and the cost associated with each edge used to connect nodes represents the level of interference at the external node. There is a constraint on the sum of these costs for the edges that are selected to be in the tree. We prove that the LCT problem is NP-complete via reduction from the vertex cover problem, and thus there are no known polynomial-time solutions to the LCT problem.

Next we formulate the LCT problem as a mixed integer linear program (MILP), which is still NP-hard, but we can use well-known computational approaches for solving small MILPs. We present a formulation that is polynomial in the number of constraints and variables, which enables input for numerical computation.

To verify the optimality of the MILP and characterize limits of maximum tree sizes, we conduct extensive Monte Carlo simulations under different scenarios. We study two cost functions: one based on the link distance to model the case where there is no information about the external node location(s), and one based on the power incident on a specific external node location. We observe the maximum tree sizes for randomly placed networks of various sizes and densities, as well as the impact of external node distance and network activity levels.

To conclude, we summarize our contributions, discuss some extensions such as other cost functions for other types of external node information. We also briefly discuss related future work on building minimum sum power backbone networks using connected dominating sets.

This page
intentionally
left blank

NETWORK SUM POWER INTERFERENCE AT AN EXTERNAL NODE

1. BACKGROUND

1.1 Introduction

As the density of wireless networked devices increases for military and commercial applications, power control becomes a critical function for the formation of ad hoc networks. Through power control, networks can limit in-network interference to increase network throughput, limit interference to nodes or devices external to the network of interest for coexistence, and reduce power and transmission range to extend network lifetime. While many studies have focused on the interference from individual nodes, it is well known that interference is more accurately modeled as the cumulative power from all active transmissions, known as the physical interference model [1], which is considered here.

The objective of this work is to connect nodes while limiting the cumulative radio frequency (RF) power emitted from the resulting network as experienced by nodes outside of the network. In particular, we focus on optimally connecting nodes to operate an ad hoc network while staying within a given threshold on the sum power experienced. The specific problem formulation is the construction of the largest connected network possible subject to the sum power constraint. In this work, we focus on centralized approaches with global knowledge of the problem, which will provide a bound on the achievable utility or cost. Developing distributed, localized algorithms will be the focus of future research.

In our approach, we represent the network as a graph where the cost associated with selecting an edge is the power incident on the external node if that link is active. We first prove that the problem of constructing the largest network subject to a sum power constraint (called the Largest Constrained Tree (LCT) problem) is NP-complete. We then formulate a mixed integer linear program (MILP) for selecting the edges and nodes in the graph that solve the LCT problem. Since integer programming is NP-hard, we use computational approaches to finding solutions to mixed integer linear programs. To characterize the optimal tree sizes and network costs, we run a range of Monte Carlo simulations for various cost models, network sizes, node densities, external node distances, and network activity.

1.2 Related Work

There have been some studies on the physical interference model in cognitive radio networks, but they typically model the average interference for some random node distribution model (e.g., Poisson point process [2, 3]). Some work only considers the spectrum/channel allocation problem without power control [4], while others consider power control with aggregate interference for the purpose of quality of service (QoS) optimization [5, 6]. In [7], the problem of downlink channel allocation and power control for maximizing the number of customers served is formulated as a mixed integer linear program. Our work differs in that we require the links be two-way and form a connected network. In this work, we study specific instances of a given node geometry and topology with an accurate physical interference model, and we apply a graph

Manuscript approved November 3, 2020.

theoretic approach to forming a connected ad hoc network. There have been extensions to the minimum spanning tree problem that introduce constraints on either node costs or edge weights [8, 9], and they have been shown to be NP-hard. The problem studied here is different in that we focus on the case where the network cannot be spanned without violating the sum power constraint, so some of the nodes will not be connected at all.

1.3 Prior Related NRL Work

NRL has a strong history of research in energy efficient ad hoc networking, which relates to the present work. The well-known Broadcast Incremental Power (BIP) and Multicast Incremental Power (MIP) algorithms were introduced in [10, 11] as a centralized algorithm for finding energy-efficient trees for broadcast and multicast. These approaches were adapted for the case with directional antennas in [12, 13]. In [14], energy efficiency was optimized over transmission probabilities for an interference channel with varying cases of available channel state information. In [15], scheduling transmissions for emptying a network of source-destination pairs was optimized for energy efficiency. We leverage our expertise in these previous efforts, specifically in constructing network trees with power/energy considerations and accurate physical interference modeling. In future work, we will likewise draw on our experience as we consider the case of directional antennas and transmission scheduling, but with the goal of limiting interference instead of energy efficiency.

2. SYSTEM MODEL

2.1 Network Model

A primary wireless network consists of nodes placed in a square deployment area. There is a receiver threshold ϕ, which is the received power required to close a link between two nodes in the network. The transmit power required to successfully transmit from node u to node v is $P_T(u,v) = \phi + pathloss(u,v)$ (dB). Here, $pathloss(u,v)$ accounts for propagation loss as well as transmit and receiver gains.

In the case where we know the location of an external node x that is not part of the network, we would like to limit the amount of power emitted from the primary network that is incident on the node. The received power at the external node for a transmission from node u to node v in the primary network is given by $P'_R(u,v,x) = P_T(u,v) - pathloss(u,x)$ (dB). The sum of the powers in both directions, which will be assigned as the cost of the link, is given by $P_R(l,x) = P'_R(u,v,x) + P'_R(v,u,x)$, where $l = (u,v)$. The total power received from some set of active links L in the primary network is given by $P_R^{tot}(L,x) = \sum_{l \in L} P_R(l,x)$. We would like to limit the total received power in two ways: 1) keep the total received power under some threshold η and 2) minimize the total received power required for a particular network topology formation.

We also include a network activity level parameter $\alpha \in (0,1]$, since not all links will be active in both directions 100% of the time. This parameter is not only a function of the data traffic load but also specific networking protocols implemented, e.g., medium access control (MAC) protocol. The network activity level can be viewed as the highest level (upper bound) of tolerable network activity, or possibly some average level, depending on how conservative the network operator needs to be or how sensitive the external node is. In either case, the total received power is given by $\alpha P_R^{tot}(L,x)$.

For additional background on modeling propagation and edge costs in the network, refer to the companion report to this paper [16].

3. LARGEST CONSTRAINED TREE PROBLEM

Given a network represented as a graph $G(V,E)$ with nodes $V = \{1,2,\ldots,n\}$ and where each activation of an undirected edge $(u,v) \in E$ incurs a cost, we would like to find the set of edges that forms the largest connected component such that the sum of the costs is below a threshold:

$$\begin{aligned} \max_{G' \subset G} \quad & |G'| \\ \text{s.t.} \quad & G' \text{ is connected} \\ & \textstyle\sum_{E(G')} \alpha w(e) \leq C \end{aligned} \tag{1}$$

where $w(e) = P_R(e,x)$ and $E(G)$ is the set of edges in graph G. We call this the Largest Constrained Tree (LCT) Problem.

3.1 Complexity

In this section, we present the theorem that describes the complexity of the LCT problem.

Theorem 1. *The Largest Weight-Constrained Tree problem is NP-complete.*

Proof. We prove this by reduction from the Vertex Cover problem:

VERTEX COVER
INSTANCE: Graph $G = (V,E)$ and positive integer $K \leq |V|$.
QUESTION: Is there a subset $V^* \subset V$ such that $|V^*| \leq K$ and, for each edge $u,v \in E$, at least one of u and v belong to V^*?

First, LCT is in NP because a nondeterministic algorithm need only guess the subgraph G' and check the constraints on the sum of the weights of the edges in the subgraph ($\leq C$), the number of vertices in the subgraph ($\leq K$), and a simple polynomial-time search (e.g., breadth-first search) can determine if the subgraph is a tree.

3.1.1 Reduction from VC

Before we transform an instance of Vertex Cover to LCT, we first create a weighted node as shown in Fig. 1, in which a node with weight N represents one node with edges connecting to $N-1$ other nodes. This is because in the largest tree construction, nodes that are connected by an edge with weight zero are included at no cost.

We transform the graph from an instance of Vertex Cover as in Fig. 2. For the new graph $G' = (V',E')$, we include in V' a node and a weighted node for each $v \in V$ as v_i and w_i, $1 \leq i \leq |V|$, respectively, a weighted node representing each $e \in E$ as e_j, $1 \leq j \leq |E|$ (called *edge nodes*), and a weighted root node r. If a vertex in the original graph is isolated with no edges connecting it, we add an edge node for it. The weight of nodes w_i are given the value m, the weight of nodes e_j are given the value n, and the weight of root node r is given the value p. Each node v_i is connected to its corresponding weighted node w_i via an edge with weight γ, and to each edge node e_j that represents an edge that it was connected to in G with weight 1. I.e.,

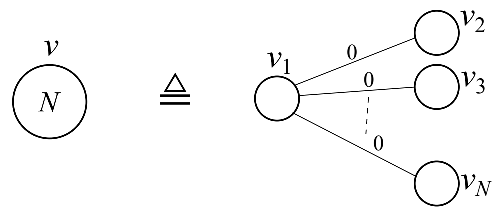


Fig. 1—Weighted node

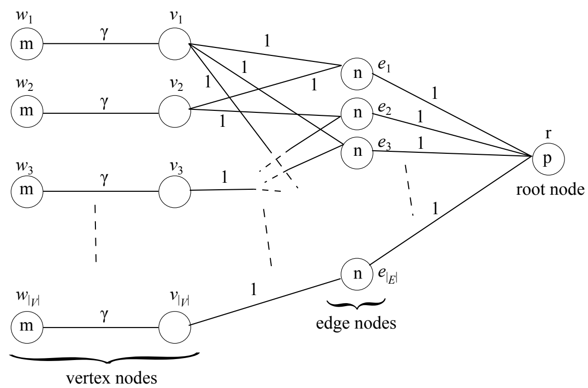


Fig. 2—Reduction from Vertex Cover to Largest Weight-Constrained Tree

if $e_k = (v,w) \in E$, then in the new graph G', there is an edge connecting v to edge node e_k, as well as w to e_k. Each of the edge nodes e_j are connected to the root node r via an edge with weight equal to 1.

We would like to build a tree in the transformed problem such that exactly K of the vertices v_i (and corresponding weighted vertices w_i, representing those in the vertex cover), all weighted edge nodes e_j (representing all the edges being covered), and the weighted root node are included in the tree. To ensure that no more than K vertices are included in the tree, we require that γ be large enough that the vertex criterion K' cannot be met by including more than K weighted vertex nodes w_i. Therefore, we require that $\gamma > 2|E|$, and $C' = \gamma K + 2|E|$. This allows for no more than K edges of weight C to be included. $|E|$ edges are to connect the edge nodes to the root node, and the other $|E|$ edges in the tree are to connect each of the edge nodes e_j to exactly one of the vertex nodes in the vertex cover.

In addition to all edge nodes w_j, we require that exactly K weighted vertex nodes w_i be included in the tree. This ensures that the edge nodes are connected by the selected vertices in the vertex cover instead of having more than $|E|$ edges from v_i to e_j. So we require $K' = (m+1)K + n|E| + p$ nodes in the tree. K' must be greater than $m(K-1) + |V| + n|E| + p$ to ensure that the vertex criterion K' cannot be satisfied by trading one of the K weighted vertex nodes w_i for the rest of the vertex nodes v_i. Therefore, $m > |V| - K$. Likewise, K' must be greater than $(m+1)K + |V| - K + n(|E|-1) + p$ to avoid trading one of the edge nodes e_j for the rest of the vertex nodes v_i. Therefore, $n > |V| - K$. Lastly, K' must be greater than $(m+1)K + |V| - K + n|E|$ to avoid trading one of the edge nodes e_j for the rest of the vertex nodes v_i. Therefore, $p > |V| - K$. Since they have the same constraint, we can choose $m = n = p = |V| - K + 1$. We have thus constructed our instance of LCT (G', C', and K') in polynomial time.

3.1.2 Solution to VC yields solution to transformed (LCT) problem and vice versa

First we argue that a solution to VC yields a solution to LCT. We have already stated that the up to $|K|$ vertex and weighted vertex nodes corresponding to nodes in the vertex cover should be included in the tree. If the number of vertices in the cover $|V'|$ is equal to K, we start by finding a matching between the v_i nodes in the cover and the edge nodes e_j, for which polynomial time algorithms exist. For each of the remaining edge nodes not in the matching, we can select one edge that connects to a vertex in the cover. If the number of vertices in the cover $|V'|$ is less than K, we first find a matching between the v_i nodes in the cover set and the edge nodes e_j and include those edges in the tree. Then we find a matching between the remaining vertex nodes and the remaining edge nodes, but we only need to include $K - |V'|$ of the edges in the tree to meet the K' criterion. We then add edges to the tree that connect the unconnected edge nodes to the vertex nodes currently in the tree. We then add edges between the vertex nodes in the tree and their corresponding weighted vertex nodes w_i. Lastly, we add the edges connecting all of the edge nodes to the root node. This yields our final solution: a subgraph G' that satisfies the C' and K' in the transformed LCT instance.

The converse, in which a solution to the transformed LCT yields a solution to the corresponding vertex cover instance, is straightforward. From the LCT solution we have exactly K vertex nodes, and by construction of the transformed tree, this set of vertices satisfies the vertex cover criterion. □

For the optimization form of LCT, we can apply a minimum spanning tree algorithm as a heuristic. Using the weighted graph with edges as constructed above, we apply a greedy-based minimum spanning tree algorithm until all nodes are connected or adding any more edges will exceed the threshold (e.g., using Prim's algorithm will be $O(|V|(|V|-1)\log|V|))$. This is solvable in polynomial time.

If the minimum spanning tree is below the threshold, then this is the optimal solution, but in general, this does not give the optimal solution. This is because the largest connected set that does not exceed the power threshold may not include the minimum power edge that we started with. So we can re-run the minimum spanning tree starting from each edge. This increases the overall complexity to $O((|V|(|V|-1))^2\log|V|)$ $(O(|V|^4\log|V|)$, which is still polynomial time solvable.

3.2 Optimization

In some integer programming formulations for minimum spanning trees, subtour elimination or cutset approaches are used in which a constraint is defined for each subset of vertices. In this case, the problem is exponential in the number of constraints and cannot practically be programmed into a software-based solver. For a formulation that is polynomial in the number of constraints and variables, we adopt the approach from [17], which is a mixed integer linear program. In this approach, a new directed graph G_{LCT} is constructed by augmenting the original graph G with two nodes $n+1$ and $n+2$, converting the undirected edges $e=(i,j)\in E$ into pairs of directed edges (i,j) and (j,i) each with the same edge cost as before $w_{ij}=w_{ji}=w(e)$, and adding zero cost directed edges from $n+1$ and $n+2$ to every node in V and $(n+1,n+2)$. The idea is to build a tree with G_{LCT} such that nodes in the connected component are connected to node $n+2$ via a root node, and the nodes not included in the component are connected directly to node $n+1$. The problem is formulated as follows:

$$
\begin{aligned}
\max \quad & \sum_{i\in V} x_i && && (2)\\
\text{s.t.} \quad & \sum_{(i,j)\in A} \alpha w_{ij} y_{ij} && \le C && (3)\\
& \sum_{i\in V} y_{n+2,i} && = 1 && (4)\\
& \sum_{i:(i,j)\in A} y_{ij} && = 1 \qquad \forall j \in V && (5)\\
& y_{n+1,i} + y_{i,j} && \le 1 \qquad \forall (i,j) \in A \setminus E' && (6)\\
& (n+1)y_{ij} + u_i - u_j && && (7)\\
& \quad +(n-1)y_{ij} && \le n \qquad \forall (i,j) \in A \setminus E' &&\\
& (n+1)y_{ij} + u_i - u_j && \le n \qquad \forall (i,j) \in E' && (8)\\
& y_{n+1,n+2} && = 1 && (9)\\
& u_{n+1} && = 0 && (10)\\
& 1 \le u_i \le n+1 && \quad i \in V \cup n+2 && (11)\\
& x_i + y_{n+1,i} && = 1 \qquad \forall i \in V && (12)\\
& x_i, y_{ij} \in \{0,1\} && \quad \forall i \in V, (i,j) \in A. && (13)
\end{aligned}
$$

Here, E' is the set of directed edges connected to $n+1$ or $n+2$, and A is the set of all directed edges (two for each edge in E, and E'). The variables x_i and y_{ij} are the decision variables for whether node i and edge (i,j) are included in the largest component. The variables u_i are used to eliminate subtours. Constraint (3) is for the sum received power, (4) is for the edge connecting $n+2$ to the root of the connected component, (5) is to ensure that all of the original nodes have exactly one neighbor, (6) ensures that if an original node is connected to another original node then it is not connected to $n+1$, (7) and (8) are the Miller-Tucker-Zemlin constraints [18] to ensure there are no subtours, (9) connects nodes $n+1$ and $n+2$, (10) and (11) is the valid range of values for the variables u_i, (12) indicates that a node is either in the tree connected to $n+2$ or directly connected to $n+1$ and not part of the connected component, and (13) defines the x_i and y_{ij} as binary decision variables. In our simulations, we solve this problem computationally using CVXPY [19], a Python package for convex optimization, and the solver used was the Cbc solver [20], which is an open-source solver that uses a branch-and-cut approach for mixed integer linear programs.

4. SIMULATION RESULTS

To validate our MILP formulation and to study the maximum average tree size for random node placement, we conduct a Monte Carlo simulation study for different network sizes, node densities, and other network characteristics. We focus on two different cost models: a distance-based cost to limit the general interference footprint without knowledge of external nodes, and an external node-based cost that measures the interference power at the external node using free space propagation. Although these costs are based on simple geometric models, the MILP formulation can accept arbitrary cost models, including precomputing complex real-world propagation models.

4.1 Distance-Based Cost

We first simulate the largest constrained tree (LCT) algorithm when the edge costs are simply based on the distance between node pairs. This distance-based cost can be applicable when the external node locations are not specified, and the goal is to just reduce the overall power emission footprint. The sum of the edges must be below $C = 100$, and the N nodes are uniformly randomly deployed in a square region with side equal to $10\sqrt{N}$. In addition to the mixed integer linear program (MILP) approaches, we also implement a brute force (BF) approach, which is an exhaustive search of all combinations of edges for the LCT.

An example of an 8 node network is shown in Fig. 3, and we see that the brute force (BF) approach and the mixed integer linear program (MILP) approach both yield the same size tree, although with different edges used. There can be multiple solutions for the LCT, and we see in this case each approach yields different trees of size 4. While there may be more edges that can be activated We have also designed the brute force approach to also choose among multiple solution the tree with lowest cost.

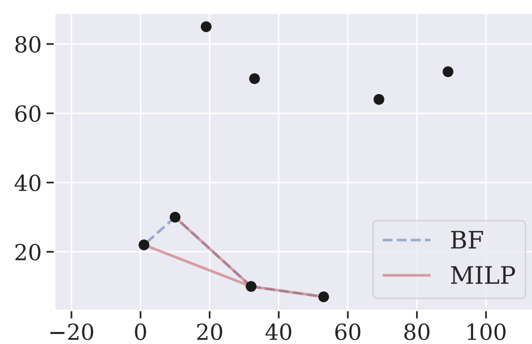
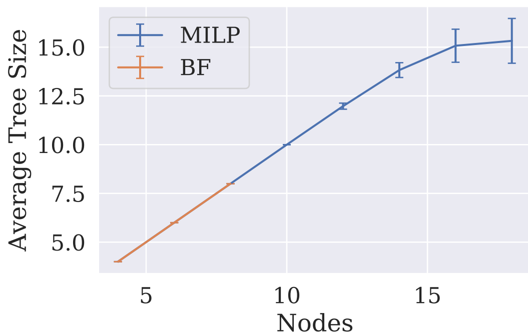


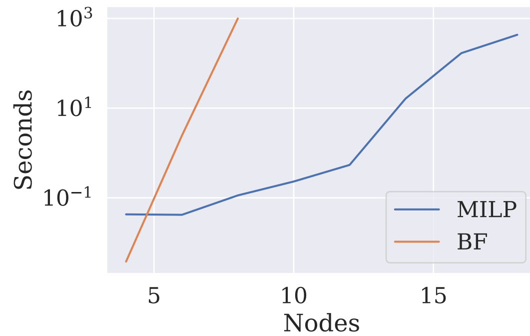
Fig. 3—8 node example, distance-based cost, $\alpha = 1$

4.1.1 Increasing Network Size

Next we have simulated the LCT algorithm for different network sizes. The cost constraint C is fixed at 100. We run both BF and MILP approaches for 50 instances of randomly placed nodes. As we increased the network size, ran each approach until the average execution time exceeded 300 seconds. Each algorithm runs for a maximum time of 1000 seconds before returning the best solution found. We first focus on the case of constant node density, where each dimension of the deployment area is scaled by $10\sqrt{N}$. The size of the LCT averaged over the 50 runs is plotted in Fig. 4(a) for various network sizes (with σ error bars). We see that for the network sizes simulated for BF (when the execution time is manageable), they yield the same size trees, which verifies that the MILP is able to find the optimal tree size. We also observe that as the number of nodes increases, so does the size of the largest tree, but it flattens out due to the fixed C. In Fig. 4(b), the average execution time of the two approaches is shown on a log scale, and we see that the execution time of the BF approach increases at a much higher rate than the MILP.



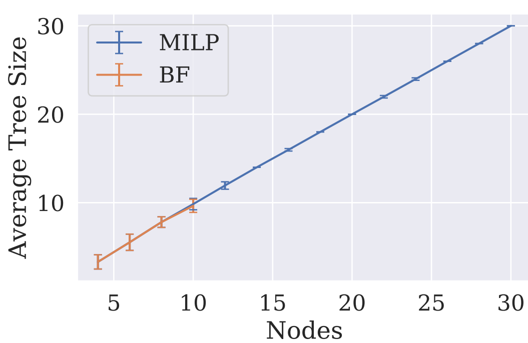
(a) LCT size vs. network size



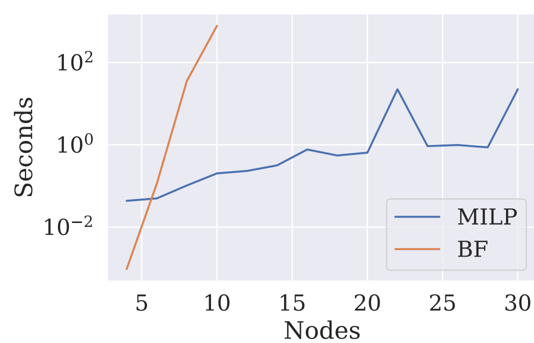
(b) Execution time vs. network size

Fig. 4—Increasing network size, fixed node density, distance-based cost, $\alpha = 1$

Next we study the case of increasing node density, where the deployment area has a side of fixed length equal to 10. The size of the LCT is plotted in Fig. 5(a) vs. the network size. Again, the BF and MILP approaches yield the same size trees, and the tree almost always contains the full network. Since the deployment area is small and fixed, the required power is almost always below the threshold. The execution time (Fig. 5(b)) for the MILP is less than the fixed node density case, likely due to the ease of finding the optimal tree size. Note that because we are averaging over only 50 instances, the curve is not completely smooth.



(a) LCT size vs. network size

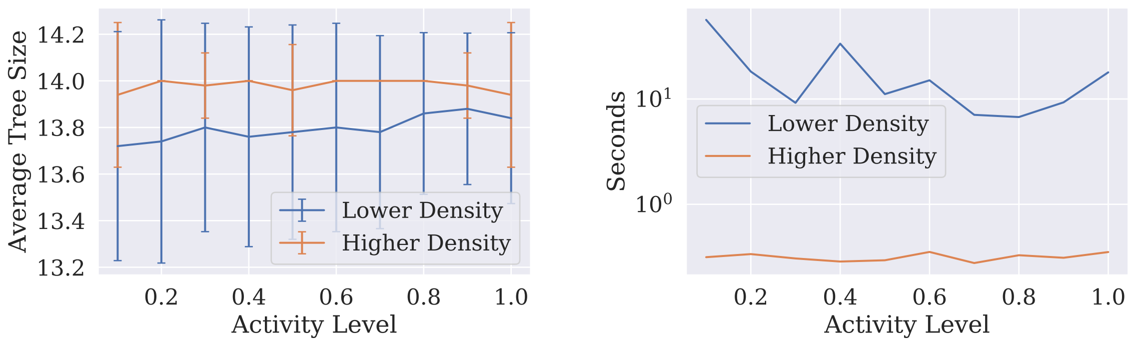


(b) Execution time vs. network size

Fig. 5—Increasing network size, increasing node density, distance-based cost, $\alpha = 1$

4.1.2 14 Node Network, Increasing Network Activity

All links in the network are unlikely to be utilized all at once, so we study a first order approximation by scaling the power by the level of network activity α. We simulate the LCT for networks of 14 nodes with different external network activity levels α. The execution time for the brute force approach for the LCT problem with 14 nodes is too unwieldy, so we omit it. We simulate a lower node density case (dimension equals $10\sqrt{N}$) and a higher node density case (dimension equals 10). We see in Fig. 6(a) that for both lower and higher density cases, the network activity level does not have a significant impact on the tree size since the distance constraint C is relaxed enough that all 14 nodes are able to be connected in most cases. The execution time is shown in Fig. 6(b), where the higher density case takes less time. In the next section, we will see examples where the network activity actually has an impact on the largest tree size.



(a) LCT size vs. network activity

(b) Execution time vs. network activity

Fig. 6—Increasing network activity, distance-based cost

4.2 External Node-Based Cost

Next we consider a cost based on the power received by an external node that is not part of the main network. We set the receiver threshold ϕ and the external node sum power threshold C equal to 5×10^{-10}. As in the distance-based case, the N nodes are uniformly randomly deployed in a square region with side equal to $10\sqrt{N}$, and the initial set of edges is limited to those between nodes that are within a range of $5\sqrt{N}$ units of each other. An example of an 8 node network is shown in Fig. 7 with the external node placed at a distance of $10\sqrt{N}$ units from the center of the original deployment area, 45 degrees from the positive x-axis. We see that the brute force (BF) approach and the mixed integer linear program (MILP) approach both yield a two node network (single edge), so this is much more sensitive than the setup in the distance-based cost case. In Fig. 8, we simulated a 64 node grid inside a 10×10 square with different locations of the external node. In Fig. 8(a), the external node is in the center and only 3 nodes could be connected. When the external node is $10\sqrt{2}$ units from the center, 45 degrees from the positive x-axis, a tree of size 8 is shown farthest away at the bottom left corner. When the external node is located 135 degrees from the positive x-axis as in Fig. 8(c), the 8-node tree is in the bottom right corner.

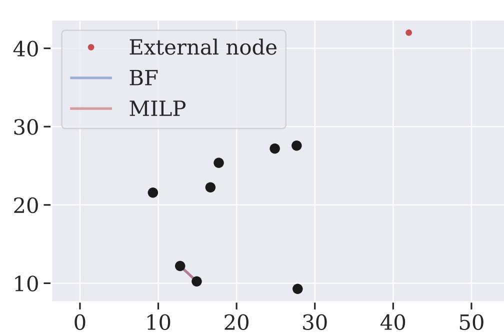
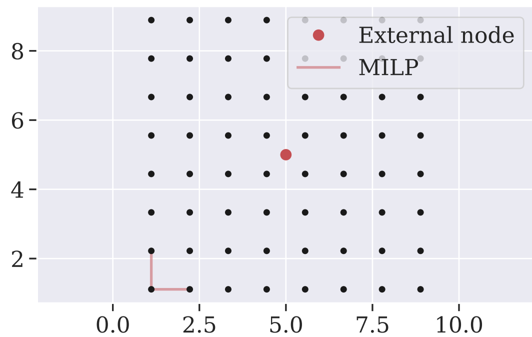
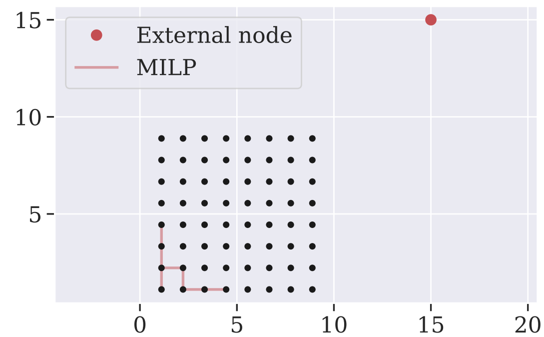


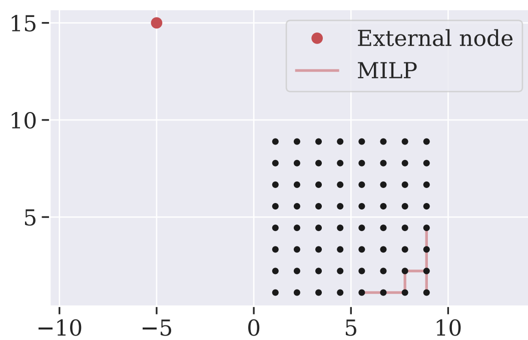
Fig. 7—8 node LCT example, $d = 1$, $\alpha = 1$



(a) $d = 0$



(b) $d = 1$



(c) $d = 1$ (northwest)

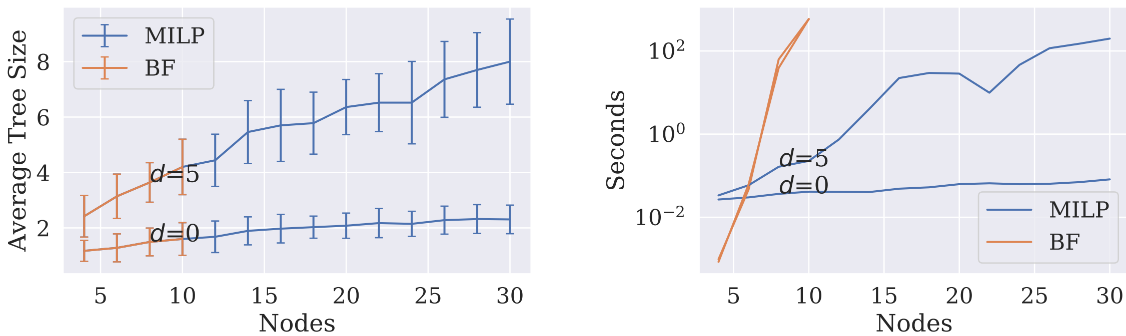
Fig. 8—64 node grid LCT example, $\alpha = 1$

4.2.1 Increasing Network Size

Next we have simulated the LCT algorithm for different network sizes, again keeping the node density constant (scaling the deployment area by $10\sqrt{N}$) and the cost constraint C fixed at 5×10^{-10}. The external node is placed 45 degrees from the original deployment area, and the distance between the center of the network deployment area and external node is $10\sqrt{2N}$. Then we run the BF and MILP approaches for 50 instances of randomly placed nodes. For each approach, as we increased the network size, we checked to see if the average execution time exceeded 300 seconds. If the 300 second threshold is exceeded, we discontinued simulating for larger network sizes, since the execution time increases exponentially for the BF approach.

The size of the LCT averaged over the 50 runs is plotted in Fig. 9(a) for various network sizes and external node distance $d = 0$ and (center of the deployment area) and $d = 5$ ($50\sqrt{N}$ from the center). We observe that for $d = 0$, the size of the largest tree is very small (< 2.5) due to the external node location being in the center and having the same sensitivity as the other nodes. As expected, it is too difficult to form a network without exceeding the interference threshold to the external node if it is in the geographic center of the deployment area. For $d = 5$, the LCT results in larger tree sizes, but still only about 40-50% of the nodes can be connected. For the execution time (Fig. 9(b)), the BF is the same for $d = 0$ and $d = 5$, but for

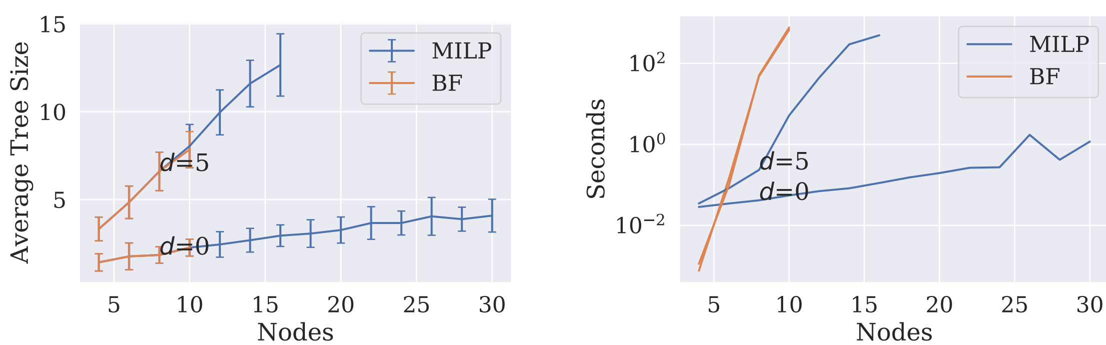
MILP the execution time for $d = 0$ is lower, likely due to the threshold being violated with very few edges selected.



(a) LCT size vs. network size (b) Execution time vs. network size

Fig. 9—Increasing network size, external node-based cost, fixed node density, $\alpha = 1$

The next simulations also look at increasing the number of nodes, but the deployment area is kept constant at 10, so the node density increases with network size. The external node is again placed 45 degrees from the original deployment area, but now the distance between the center of the network deployment area and external node is $10d\sqrt{2}$. Compared to the fixed density case, the LCT results in larger tree sizes (Fig. 10(a)), and for $d = 5$, it is close to including the full network. Unfortunately, for $d = 5$, the execution time is too long to evaluate for network sizes greater than $N = 16$ (Fig. 10(b)).

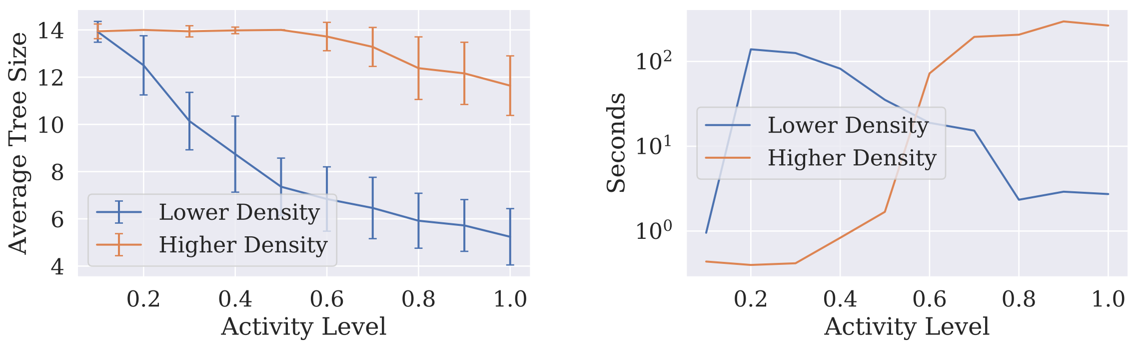


(a) LCT size vs. network size (b) Execution time vs. network size

Fig. 10—Increasing network size, external node-based cost, increasing node density, $\alpha = 1$

4.2.2 14 Node Network, Increasing Network Activity

As with the distance-based cost, we simulate the LCT for networks of 14 nodes with different external network activity levels α. We omit the brute force approach due to the long execution time for 14 nodes. We simulate a lower node density case (dimension equals $10\sqrt{N}$, external node distance is $10d\sqrt{2N}$) and a higher node density case (dimension equals 10, external node distance is $10d\sqrt{2}$) for $d = 5$. The largest tree size is plotted for both cases in Fig. 11(a). For the lower density case, we observe that all nodes are in the tree at $\alpha = 0.1$ and gradually drops to about half the size at $\alpha = 0.5$. For the higher density case, we observe that the largest tree size includes all nodes for $\alpha \leq 0.5$. We observe in Fig. 11(b) some interesting behavior of the execution time for the lower density case. It starts low because it is easy to connect all nodes within the threshold, but at $\alpha = 0.2$, the execution time increases dramatically as the constraint starts excluding some nodes from connecting. As α further increases, the execution time decreases as the constraint is met sooner with fewer nodes connected.



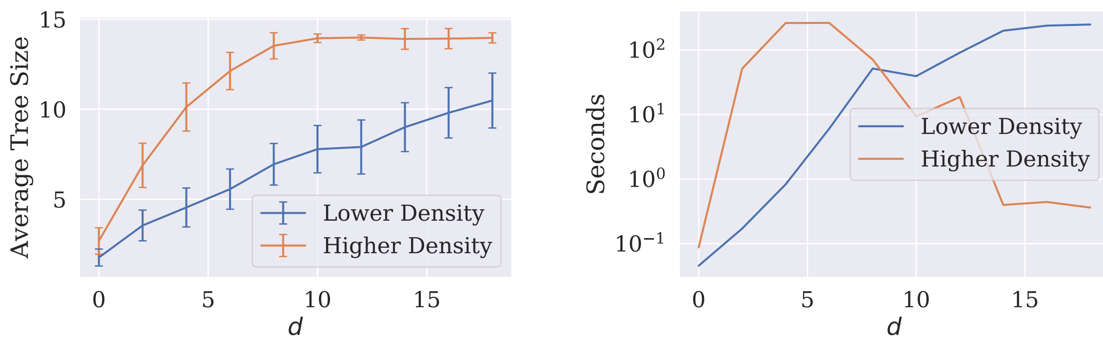
(a) LCT size vs. network activity

(b) Execution time vs. network activity

Fig. 11—Increasing network activity, distance-based cost

4.2.3 14 Node Network, Increasing External Node Distance

To understand the impact of the distance of the external node on the network connectivity and power emitted, we fix the network size at 14 nodes and simulate the LCT algorithm for different external node distances d. Again, we omit the brute force approach for the LCT problem with 14 nodes. We consider both lower density (dimension equals $10\sqrt{N}$) and higher density cases (dimension equals 10). For the lower density case, the tree size is steadily increasing as the external node gets farther away, but it is stopped short due to the execution time being too long. For the higher density case, it includes all 14 nodes for $d \geq 10$. For the execution time shown in Fig. 12(b), we observe that the execution time is small when the whole network is connected or when there are few nodes to connect, as noted in the previous section (Fig. 11(b)).



(a) LCT size vs. external node distance (b) Execution time vs. external node distance

Fig. 12—Increasing external node distance, external node-based cost, $\alpha = 1$

5. CONCLUSION AND FUTURE WORK

This work has studied the problem of constructing the largest tree subject to a constraint on the sum power interference experienced by a node outside of the network. We have proven that the LCT problem as formulated is NP-complete, and we presented a MILP formulation that can be solved using computational methods for relatively small network sizes, but can handle larger networks than a brute force method. Simulations characterize the average maximum tree size for various network scenarios, and increasing the node density or reducing network activity are ways to increase the tree size within the sum power constraint.

There are a number of extensions to this work under consideration. First, a more efficient heuristic algorithm is needed to handle larger numbers of nodes. It is also important to develop distributed approaches to discovering these tree topologies without violating the interference constraint. We can also study different ways of modeling the cost when there are multiple external nodes or when only a probability distribution of nodes is known. The problem can also be extended to consider heterogeneity in the network, in which directional transmission or other modalities such as free space optical communication can be utilized. More significant reformulations of the problem include multi-objective optimization (e.g., throughput and tree size) and asymmetric routing (directed trees from source to destinations).

Another related line of research under investigation involves constructing a backbone network with minimum sum power, under some constraint on the of leaf nodes. We formulate this as a connected dominating set problem with minimum cost, and plan to use a similar graph theoretic and MILP formulation to find an optimal solution.

REFERENCES

1. P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory* **46**(2), 388–404 (2000).

2. M. Timmers, S. Pollin, A. Dejonghe, A. Bahai, L. Van der Perre, and F. Catthoor, "Accumulative Interference Modeling for Distributed Cognitive Radio Networks," *JCM* **4**(3), 175–185 (2009).

3. M. Pirmoradian and C. Politis, "Cumulative interference," *IEEE Vehicular Technology Magazine* **5**(3), 39–45 (2010).

4. Y. Li, L. Zhao, C. Wang, A. Daneshmand, and Q. Hu, "Aggregation-based spectrum allocation algorithm in cognitive radio networks," Proceedings of the 2012 IEEE Network Operations and Management Symposium, 2012, pp. 506–509.

5. T. Zheng, Y. Qin, D. Gao, J. Duan, and H. Zhang, "Hybrid Model Design and Transmission Rate Optimize with Interference Temperature Constraints in Cognitive Radio Sensor Networks," Proceedings of the 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing, 2011, pp. 1–4.

6. S. Gong, P. Wang, Y. Liu, and W. Zhuang, "Robust Power Control with Distribution Uncertainty in Cognitive Radio Networks," *IEEE Journal on Selected Areas in Communications* **31**(11), 2397–2408 (2013).

7. A. T. Hoang and Y. Liang, "Downlink Channel Assignment and Power Control for Cognitive Radio Networks," *IEEE Transactions on Wireless Communications* **7**(8), 3106–3117 (2008).

8. R. Ravi and M. X. Goemans, "The constrained minimum spanning tree problem," Proceedings of the Scandinavian Workshop on Algorithm Theory (Springer), 1996, pp. 66–75.

9. V. Aggarwal, Y. P. Aneja, and K. Nair, "Minimal spanning tree subject to a side constraint," *Computers & Operations Research* **9**(4), 287–296 (1982).

10. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," Proceedings of the Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), volume 2, 2000, pp. 585–594 vol.2.

11. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Algorithms for energy-efficient multicasting in static ad hoc wireless networks," *Mobile Networks and Applications* **6**(3), 251–263 (2001).

12. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-limited wireless networking with directional antennas: the case of session-based multicasting," Proceedings of the Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, volume 1, 2002, pp. 190–199 vol.1.

13. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-aware wireless networking with directional antennas: the case of session-based broadcasting and multicasting," *IEEE Transactions on Mobile Computing* **1**(3), 176–191 (2002).

14. G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier, and A. Ephremides, "Impact of channel state information on energy efficient transmission in interference channels," Proceedings of the 2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014, pp. 474–481.

15. G. D. Nguyen, S. Kompella, C. Kam, J. E. Wieselthier, and A. Ephremides, "Minimum-energy link scheduling for emptying wireless networks," Proceedings of the 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015, pp. 207–212.

16. J. P. Macker, C. Bowers, S. Kompella, and C. Kam, "Wireless Ad Hoc Network Topology Control with Link Constraints," NRL/FR/5522–20-10,404, U.S. Naval Research Laboratory, Washington, DC, 2020.

17. N. Fan and J. P. Watson, "Solving the Connected Dominating Set Problem and Power Dominating Set Problem by Integer Programming," *Lecture Notes in Computer Science* pp. 371–383 (2012), ISSN 1611-3349, doi:10.1007/978-3-642-31770-5_33. URL `http://dx.doi.org/10.1007/978-3-642-31770-5_33`.

18. C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer Programming Formulation of Traveling Salesman Problems," *Journal of the ACM (JACM)* **7**(4), 326–329 (Oct 1960), ISSN 1557-735X, doi:10.1145/321043.321046. URL `http://dx.doi.org/10.1145/321043.321046`.

19. "CVXPY Website." URL `https://www.cvxpy.org/`.

20. "Github coin-or branch-and-cut (cbc) solver website." URL `https://github.com/coin-or/Cbc`.