Information Processing Letters 110 (2010) 774–777

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Multicasting in the hypercube, chord and binomial graphs

Christopher C. Cipriano, Teofilo F. Gonzalez *

Department of Computer Science, University of California, Santa Barbara, CA 93106, United States

ARTICLE INFO

Article history:
Received 12 March 2009
Received in revised form 11 June 2010
Accepted 21 June 2010
Available online 1 July 2010
Communicated by F.Y.L. Chin

Keywords:
Algorithms
NP-Complete
n-cube
Binomial graphs
Chord
Multicasting
Steiner trees

ABSTRACT

We discuss multicasting for the *n*-cube network and its close variants, the Chord and the Binomial Graph (BNG) Network. We present *simple* transformations and proofs that establish that the sp-multicast (shortest path) and Steiner tree problems for the *n*-cube, Chord and the BNG network are NP-Complete, even when every destination vertex is at a distance two from the source vertex.

1. Introduction

Multicasting is a communication primitive that allows a (source) vertex in a network to send a message to multiple destination vertices. In this paper we consider regular networks without edge weights (or costs). The communication steps are modeled by a *tree*. A tree connects (directly or indirectly) the source vertex to all the destination vertices, and may include other vertices in the network. There are many different multicast trees and objective functions. The first type of tree has the minimum number of edges (links). The problem of generating this type of tree is known as the *minimum Steiner tree* (*MST*) problem.[1] The second type of tree has the minimum number of edges (links) provided that every path from the source vertex to a destination vertex is a shortest path in the original network. We refer to this problem as the *shortest path multicast*(*sp-multicast*) problem.

The decision version of these problems are formally defined below. The Steiner tree (ST) decision problem is: given an undirected graph $G=(V,E)$, a subset of vertices, $K=\{u_0,u_1,\ldots,u_k\}\subseteq V$, and a positive integer r, is there a sub-tree $T=(V_T,E_T)$ of G ($V_T\subseteq V$ and $E_T\subseteq E$) such that (a) $K\subseteq V_T$, and (b) the number of edges in E_T is at most r?

The sp-multicast tree decision problem is the Steiner tree decision problem with the added constraint $d_T(u_0,u_i)=d_G(u_0,u_i)$ for $1\leqslant i\leqslant k$, where $d_T(a,b)$ and $d_G(a,b)$ is the number of edges in a shortest path from a to b in T and G, respectively.

We study these problems in the context of the *n-cube*. An *n-cube* (*hypercube*) consists of 2^n vertices or processors. Every vertex in the *n*-cube is represented by an *n*-bit string and there is an edge between two vertices if their bit representations disagree in exactly one bit. For the *n*-cube graph, we refer to the above problems as the *n*-cube Steiner tree problem and the *n*-cube sp-multicast tree problem. An instance of the *n*-cube Steiner tree or *n*-cube sp-multicast decision consists of $k+1$ vertices and an integer r. Since the structure of the *n*-cube is uniform,

* Corresponding author.
E-mail addresses: ccc@cs.ucsb.edu (C.C. Cipriano), teo@cs.ucsb.edu (T.F. Gonzalez).

[1] Traditionally Steiner tree problems are defined for weighted graphs with the objective being to minimize the total weight of the edges in the tree.

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2010.06.009

one does not need to provide the vertices and edges of the n-cube as part of the input. Each of the destination vertices is specified by an n-bit string and r is an $(n + \log n)$-bit binary number. Note that even when the input size is bounded above by a polynomial on n, the number of vertices and edges of the n-cube is exponential on n.

Graham and Foulds [1] studied the MST problem for the n-cube in order to determine the possibility of computing specific biological sciences problems in reasonable time. Their work resulted in a complex proof for the NP-Completeness of the decision version of the n-cube Steiner tree problem. Later on, a complex transformation and proof was used to establish that the n-cube sp-multicast problem is NP-Complete [2]. In this paper we present *one simple* transformation and proofs that establish the NP-Completeness of these two problems. We establish that these problems are NP-Complete even when every destination vertex is at a distance two from the source vertex.

A generalization of the n-cube is the binomial graph network and the n-Chord. The binomial graph network provides desirable topological properties in terms of scalability and fault-tolerance [3] and the n-Chord has been used for structured peer-to-peer (P2P) networks [3]. Formally, a *BiNomial Graph* (or *n-BNG*) network consists of n vertices. The vertices are denoted $\{0, 1, \ldots, n-1\}$. Let k be the largest integer such that $2^k \leqslant n-1$. Every vertex i in the n-BNG network has (clockwise) edges to vertices $\{(i+2^0) \bmod n, (i+2^1) \bmod n, \ldots, (i+2^k) \bmod n\}$ and (counterclockwise) edges to vertices $\{(i-2^0) \bmod n, (i-2^1) \bmod n, \ldots, (i-2^k) \bmod n\}$. The n-BNG network is referred to as the *k-Chord* (or simply the *Chord*) when $n = 2^k$ for some integer $k \geqslant 1$.

It is simple to show that deleting some edges from an n-Chord results in an n-cube. Therefore, message communication in the n-Chord can be performed more efficiently than in the n-cube, but the number of edges (links) in the n-Chord is twice the number of edges in the n-cube and therefore more expensive to deploy. The BNG network has properties similar to the Chord.

There is a trivial algorithm to implement optimum unicasting (multicasting to one destination) in the n-cube. Optimal polynomial time algorithms for unicasting have been developed for both the Chord and the binomial graph network [4,3]; however, there has not been a lot of work on multicasting in these topologies. It was conjectured that optimum sp-multicast trees for the binomial graph network can be constructed by simply using the unicast algorithm from the source to all destinations while choosing intermediate vertices that decrease network traffic [3]. While this explanation does describe a procedure to construct minimum sp-multicast trees, there is no known polynomial time algorithm that can implement it because there is no known efficient algorithm to choose intermediate vertices that decrease network traffic. We prove that no such polynomial time implementation exists if $P \neq NP$. In this paper we present proofs of NP-Completeness for the MST and sp-multicast tree for the Chord and the BNG by simple modifications to NP-Completeness proofs for the corresponding problems defined over the n-cube. We establish that these problems are NP-Complete even when every destination vertex is at a distance two from the source vertex.

2. NP-Completeness results

To establish our NP-Completeness results we use the vertex cover problem. The Vertex Cover (VC) decision problem is: given an undirected graph $G = (V = \{1, 2, \ldots, n\}, E)$ and an integer c, is there a vertex cover V' with cardinality at most c, i.e., a set of vertices V' such that $V' \subseteq V$ and every edge $e \in E$ is incident upon at least one vertex in V'?

Theorem 2.1. *The n-cube sp-multicast tree decision problem is NP-Complete even when every vertex in $K/\{u_0\}$ is at a distance two from the source vertex u_0.*

Proof. Our polynomial time transformation from the VC decision problem is defined as follows. Let $G = (V, E)$, an undirected graph, and c, a positive integer, be any instance of the VC decision problem. Let $n = |V|$ and $m = |E|$. We construct the instance $(K = \{u_0, u_1, \ldots, u_k\}, r)$ of the n-cube sp-multicast tree decision problem as follows. The vertex u_0 is the vertex in the n-cube represented by the string of n 0-bits. For every edge $e_l = \{i, j\}$ in G we define the vertex u_l in the n-cube represented by the string of n 0-bits except for two bits that are 1-bits at positions i and j. Clearly, $k = m$ and let $r = c + k$.

We now prove our transformation is correct. Let integers $i_1, i_2, \ldots, i_c$ represent the vertices in a vertex cover with cardinality c for G. Now lets define the set of vertices $\{j_1, j_2, \ldots, j_c\}$. Vertex j_l (in the n-cube) is represented by the string of n 0-bits except for a 1-bit at position i_l. Since every edge e_l is incident to at least one vertex in $\{i_1, i_2, \ldots, i_c\}$, then vertex u_l is a neighbor of at least one vertex in $\{j_1, j_2, \ldots, j_c\}$ in the n-cube. Define the sp-multicast tree MT by the set of vertices $K \cup \{j_1, j_2, \ldots, j_c\}$ and the set of edges of the form $\{u_0, j_i\}$ plus one edge from each vertex u_l to a vertex in $\{j_1, j_2, \ldots, j_c\}$. These edges exist as $\{i_1, i_2, \ldots, i_c\}$ is a vertex cover for G. The number of edges in the tree is $r = k + c$. Therefore, (K, r) has an sp-multicast tree with at most r edges.

Conversely, let T be an sp-multicast tree with at most r edges for the instance (K, r). Clearly we may assume that all the edges join a vertex with exactly one 1-bit to either a vertex with zero 1-bits (vertex u_0), or a vertex with exactly two 1-bits (u_l vertex). Therefore every vertex in $\{u_1, u_2, \ldots, u_k\}$ has an edge to a vertex in the n-cube with exactly one 1-bit for a total of k edges. Let $\{j_1, j_2, \ldots, j_f\}$ be the vertices with exactly one 1-bit in T. All of these vertices are neighbors of u_0 in the n-cube, so the f edges to join them to u_0 must be in T. In order for the tree to have at most r edges, it must be that $f \leqslant c$. For $1 \leqslant l \leqslant f$ define $i_l = b$, where j_l has its 1-bit at position b. Clearly, the set of vertices $\{i_1, i_2, \ldots, i_f\}$ is a vertex cover for G with cardinality at most c. □

Before we establish that the n-cube Steiner tree decision problem is NP-Complete, we establish the following two technical lemmas.

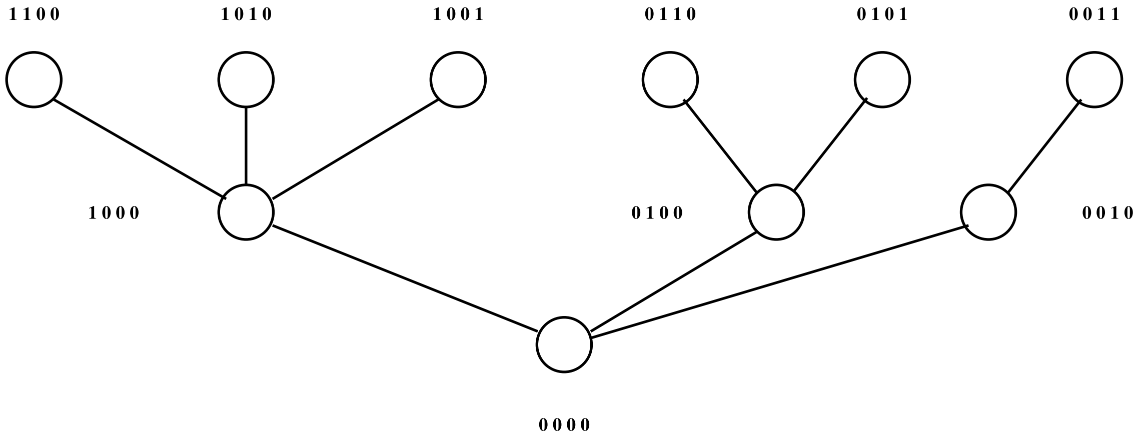

Fig. 1. An sp-multicast tree for u_0 and six vertices each with two 1-bits in two of four possible positions.

Lemma 2.1. *Let y be a vertex of the n-cube with exactly three 1-bits and let Γ be a non-empty subset of vertices each with exactly two 1-bits that are neighbors of y. Let γ be the number of vertices in Γ. Then, $1 \leqslant \gamma \leqslant 3$, and there is an sp-multicast tree rooted at $u_0 = 00\ldots0$ with at most $\lfloor 4\gamma/3 \rfloor + 1$ edges that includes all the vertices in Γ.*

Proof. The proof is left as an exercise to the reader. □

Lemma 2.2. *Let x be a vertex of the n-cube with exactly four 1-bits and let Γ be a non-empty subset of vertices each with exactly two 1-bits which are in common with the 1-bits in x. Let γ be the number of vertices in Γ. Then, $1 \leqslant \gamma \leqslant 6$ and there is an sp-multicast tree rooted at u_0 that includes all the vertices in Γ with at most $\gamma + 3$ edges.*

Proof. Assume without loss of generality that $n = 4$. Fig. 1 shows an sp-multicast tree for the case when γ equals to 6 with $\gamma + 3$ edges. When γ is less than six just delete from Fig. 1 the destination vertices that are not in Γ as well as superfluous vertices and edges, and the resulting multicast tree has at most $\gamma + 3$ edges. □

Theorem 2.2. *The n-cube Steiner tree decision problem is NP-Complete even when every vertex in $K/\{u_0\}$ is at a distance two from the source vertex u_0.*

Proof. Our polynomial time transformation is the same one as the one used in Theorem 2.1. To establish that this is a valid transformation we use the proof of Theorem 2.1 and prove that if there is a Steiner tree with at most r edges, then there is also an sp-multicast tree with at most r edges.

Let $f(I)$ be any problem instance generated by the polynomial transformation. Let ST be a Steiner tree with at most r edges that is not an sp-multicast tree. Assume without loss of generality that when viewing the tree ST as a tree rooted at u_0, all of its leaves are elements of the set $\{u_1, u_2, \ldots, u_k\}$. The standard parent-child relationship is defined between neighbor vertices when viewing the tree as rooted at u_0. Assume without loss of generality that every vertex in ST consisting of exactly one 1-bit is a child of u_0. If this were not the case, the following simple transformation can be applied to alter the tree to satisfy this condition. Let a be a one 1-bit vertex in ST that is not a child of u_0, and let b be the parent of a. In ST delete the edge $\{a, b\}$ and add the edge $\{u_0, a\}$. We now show that instance $f(I)$ has an sp-multicast tree with at most r edges. Let $sp(ST)$ be the number of vertices in $\{u_1, u_2, \ldots, u_k\}$ that have a path in ST to u_0 with exactly two edges. Clearly $sp(ST) < k$. Our approach is to show that ST can be transformed into another Steiner tree ST' with at most r edges such that $sp(ST') > sp(ST)$. After applying this argument at most k times we know that instance $f(I)$ has an sp-multicast tree with at most r edges.

Let $u \in \{u_1, u_2, \ldots, u_k\}$ be a vertex whose (simple) path in ST from u_0 to u is the longest. Let P be the path in ST that starts at u_0 and ends at u. Clearly, path P has at least four edges. Let w, x and y be the last three vertices just before u in path P, i.e., the path from u_0 to u visits vertex w, then it is followed by the edges $\{w, x\}$, $\{x, y\}$ and $\{y, u\}$, to reach vertices x, y and u in that order.

By our assumptions the number of 1-bits of y must be equal to three. The sub-tree ST_w is defined as ST after deleting all the sub-paths originating at vertex u_0 that do not include vertex w. It is convenient to visualize ST_w as a tree rooted at w. All the neighbors of w are said to be the children of w in ST_w. Define the parent-child relationship for the vertices in ST_w when viewing the tree ST_w rooted at w. Every leaf in ST_w is at a distance at most three from w and it is a vertex in $\{u_1, u_2, \ldots, u_k\}$. There are two cases depending on the number of 1-bits of x.

Case 1. The number of 1-bits of x is two. Let α be the number of children of y in ST_w. Since the parent of y, that is x, and all the children of y in ST_w have exactly two 1-bits and y has three 1-bits, it must be that $1 \leqslant \alpha \leqslant 2$. By Lemma 2.1 we know there is an sp-multicast tree rooted at u_0, which we call A, that includes all the children of y in ST_w with $\alpha + 1$ edges (as $\alpha \leqslant 2$). Define ST' as ST after deleting vertex y and the edges incident to vertex y, and adding the vertices and edges in sp-multicast tree A that are not in ST.

Case 2. The number of 1-bits of x is four. Let β be the number of children of x in ST_w Let α be the number of vertices in $\{u_1, u_2, \ldots, u_k\}$ that are descendants of x in the sub-tree ST_w. If $\beta = 1$, then α is at most three. By Lemma 2.1 we know there is an sp-multicast tree rooted at u_0, which we call A, that includes all the children of y in ST_w with $\alpha + 2$ edges. Define ST' as ST after deleting vertices x and y, as well as all the edges incident to them, and adding the vertices and edges in sp-multicast tree A that are not in ST. This transformation deletes $\alpha + 2$ edges and adds at most $\alpha + 2$ edges. On the other hand if $\beta > 1$, then by Lemma 2.2 we know there is an sp-multicast tree rooted at u_0, which we call A, that includes all the leaves that are descendants of x in ST_w with $\alpha + 3$ edges, as x has four 1-bits. Define ST' as ST after deleting vertex x, all the children of x, as well as all the edges incident to them, and adding the vertices and edges in the sp-multicast tree A that are not in ST. The above transformation deletes $\alpha + \beta + 1$ edges and adds at most $\alpha + 3$ edges. Since $\beta > 1$ we know that $\alpha + \beta + 1 \geqslant \alpha + 3$.

In all cases ST' does not have more edges than ST and $sp(ST') > sp(ST)$. Eventually $sp(ST')$ will be equal to k and ST' will be an sp-multicast tree with at most r edges. This concludes the proof of the theorem. □

We now establish that the corresponding problems on the Chord are NP-Complete.

Theorem 2.3. *The Chord sp-multicast and Steiner tree decision problems are NP-Complete even when every vertex in $K/\{u_0\}$ is at a distance two from the source vertex u_0.*

Proof. The reductions are similar to the ones in the previous theorems. The difference is that between every pair of bits of the vertices in K in the previous reduction, which we call *box bits*, we add a bit pattern called the *signature*.

Let $G = (V, E)$, an undirected graph, and c, a positive integer, be any instance of the VC decision problem. Let $n = |V|$ and $m = |E|$. We construct the instance $(K = \{u_0, u_1, \ldots, u_k\}, r)$ of the t-Chord multicast tree decision problem as follows, where $k = m$, $r = c + m$, and $t = n + (n - 1) * (2r + 3)$. Every vertex in K in our reduction consists of n box bits and $n - 1$ signatures arranged in the order $b, s, b, s \ldots, b, s, b$, where b is a box bit, and s is the signature. The signature is the $2r + 3$ bit pattern $0101 \ldots 010$. Vertex u_0 in the t-Chord has all the box bits equal to zero. For every edge $e_l = \{i, j\}$ in G we define the vertex u_l in the t-Chord with all the box bits equal to zero, except for the ith and jth box bits which are 1-bits. Therefore, k is equal to m.

The proof that the transformation is correct is based on the proofs of Theorems 2.1 and 2.2, and the argument that no two neighbors in a tree with r edges have two or more different box bits. The reason for this is that in order for two neighbors to have two or more different box bits at least one of the signatures must equal to all zeros or all ones. But each signature has $r + 1$ 1-bit runs and at each step one can reduce the number of 1-bit runs in a signature by at most one. Since the whole tree has at most r edges, transforming one signature into all ones or all zeros is not possible as this would take a tree with more than r edges. □

3. Discussion

We presented simple proofs to establish that the Steiner and sp-multicast tree decision problems on the n-cube, Chord and BNG networks are NP-Complete. Our reductions and the ones in [1,2] define problem instances where the number of bits to represent the nodes in the n-cube is proportional to the size n of the NP-Complete problem being reduced. However this implies that the n-cube has 2^n vertices, though n vertices (n^2 bits) is the input to the n-cube problem. An important open problem is to determine whether or not our problems remain NP-Complete when the reduction is for a hypercube with $O(P(n))$ vertices, where $P(n)$ is a polynomial on n. Gonzalez and Serena [5] have shown that some problems defined over the hypercube are NP-Complete even under this condition.

Acknowledgements

We like to thank an anonymous referee for all his/her comments and suggestions to improve the readability of this paper.

References

[1] R.L. Graham, L.R. Foulds, Unlikelihood that minimal phylogenies for a realistic biological study can be constructed in reasonable computation time, Mathematical Biosciences 60 (1982) 133–142.

[2] H.-A. Choi, A.H. Esfahanian, On complexity of a message-routing strategy for multicomputer systems, in: R.H. Mohring (Ed.), Lecture Notes in Computer Science, vol. 484, Springer-Verlag, 1991, pp. 170–181.

[3] T. Angkun, G. Bosilca, B. Vander Zanden, J. Dongarra, Optimal routing in binomial graph networks, in: Proceedings of the Eight International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'07), IEEE Computer Society, 2007, pp. 363–369.

[4] P. Ganesan, G.S. Manku, Optimal routing in chord, in: Proceedings of the Fifteenth Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'04), Society for Industrial and Applied Mathematics, 2004, pp. 176–185.

[5] T.F. Gonzalez, D. Serena, Pairwise disjoint shortest paths in the n-cube and related problems, Theoretical Computer Science 369 (1) (2006) 427–435.