Embeddings Between Circulant Networks and Hypertrees

A. Arul Shantrinal, R. Sundara Rajan*, T.M. Rajalaxmi and K. Jagadeesh Kumar

Abstract—Graph embedding has been known as a powerful tool for implementation of parallel algorithms and simulation of different interconnection networks. In this paper, we compute the exact wirelength of embedding circulants network into hypertrees and vice-versa.

Index Terms—Embedding, congestion, wirelength, circulant network, chord graph, hypertree

I. INTRODUCTION AND TERMINOLOGY

INTERCONNECTION networks play an important role in parallel computing systems. An interconnection network can be represented by a graph \( G = (V, E) \), where \( V \) represents the node set and \( E \) represents the edge set. In this paper, we use graphs and interconnection networks (networks for short) interchangeably. Graph embedding is a technique in parallel computing that maps a guest graph into a host graph (usually an interconnection network). There are many applications of graph embedding, such as architecture simulation, processor allocation, VLSI chip design, etc. Architecture simulation is the simulation of one architecture by another. This can be modeled as a graph embedding, which embeds the guest architecture into the host architecture, where the nodes of the graph represent the processors and the edges of the graph represent the communication links between the processors. In parallel computing, a large process is often decomposed into a set of small sub processes that can execute in parallel with communications among these sub processes. According to these communication relations among these sub processes, a graph can be obtained, in which the nodes in the graph represent the sub processes and the edges of the graph represent the communication links between these sub processes. Thus, the problem of allocating these sub processes into a parallel computing systems can be again modeled as a graph embedding problem. The problem of laying out circuits on VLSI chips can also be reduced to graph embedding problems [1].

The quality of an embedding can be measured by certain cost criteria. One of these criteria which is considered very often is the wirelength. The wirelength of an embedding is the sum of the dilations in host graph of edges in guest graph. The wirelength of a graph embedding arises from VLSI designs, data structures and data representations, networks for parallel computer systems, biological models that deal with cloning and visual stimuli, parallel architecture, structural engineering and so on [2, 3] Graph embeddings have been well studied for a number of networks [4 – 11].

The circulant network is a natural generalization of the double loop network [12]. Circulant graphs have been used for decades in the design of computer and telecommunication networks due to its optimal fault-tolerance and routing capabilities [13]. They are also used in VLSI design and distributed computation [14 – 16] Circulant graphs have been employed for designing binary codes [17]. Theoretical properties of circulant graphs have been studied extensively and surveyed by Bermond et al. [14]. Every circulant graph is a Cayley graph, and is therefore vertex transitive [2]. Most of the earlier research concentrated on using the circulant graphs to build interconnection networks for distributed and parallel systems [13, 14].

An overlay network is a computer network which is built on the top of another network. Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network. For example, distributed systems such as cloud computing, peer-to-peer networks, and client-server applications are overlay networks. Chord graphs introduced by Stoica et al. [18], are a structured peer-to-peer architecture based on distributed hash tables (DHTs) [19]. In [6], the chord graph is considered as an overlay network. The rest of the paper is organized as follows: Section 2 gives definitions and other preliminaries. In Section 3, we determine the wirelength of embedding circulant network into hypertree and vice-versa. In Section 4, we discuss the time complexity of the wirelength. Finally, concluding remarks and future works are given in Section 5.

II. BASIC CONCEPTS

In this section we give the basic definitions and preliminaries related to embedding problems.

Definition 1. [11] Let \( G \) and \( H \) be finite graphs. An embedding of \( G \) into \( H \) is a pair \((f, P_f)\) defined as follows:
1) \( f \) is a one-to-one map from \( V(G) \rightarrow V(H) \)
2) \( P_f \) is a one-to-one map from \( E(G) \) to \( \{P_f(uv) : P_f(uv) \text{ is a path in } H \text{ between } f(u) \text{ and } f(v) \text{ for } (uv) \in E(G) \} \).
For brevity, we denote the pair \((f, P_f)\) as \(f\).

**Definition 2.** [11] Let \(f : G \to H\) be an embedding. For \(e \in E(H)\), let \(EC_f(e)\) denote the number of edges \((uv)\) of \(G\) such that \(e\) is in the path \(P_f(uv)\) between \(f(u)\) and \(f(v)\) in \(H\). In other words,

\[
EC_f(e) = |\{(uv) \in E(G) : e \in P_f(uv)\}|.
\]

Then the edge congestion of \(f : G \to H\) is \(EC_f(G, H) = \max EC_f(e)\), where the maximum is taken over all edges \(e\) of \(H\). The edge congestion of \(G\) into \(H\) is defined as \(EC(f, G, H) = \min EC_f(G, H)\), where the minimum is taken over all embeddings \(f : G \to H\).

On the other hand, if \(S\) is any subset of \(E(H)\), then \(EC_f(S) = \sum_{e \in S} EC_f(e)\).

If we think of \(G\) as representing the wiring diagram of an electronic circuit, with the vertices representing components and the edges representing wires connecting them, then the edge congestion \(EC(G, H)\) is the minimum, over all embeddings \(f : V(G) \to V(H)\), of the maximum number of wires that cross any edge of \(H\) [20], see Figure 1.

**Definition 3.** [8] The wirelength of an embedding \(f\) of \(G\) into \(H\) is given by

\[
WL_f(G, H) = \sum_{(u, v) \in E(G)} d_H(P_f(uv)) = \sum_{e \in E(H)} EC_f(e)
\]

where \(d_H(P_f(uv))\) denotes the length of the path \(P_f(uv)\) in \(H\). The wirelength of \(G\) into \(H\) is defined as

\[
WL(G, H) = \min WL_f(G, H)
\]

where the minimum is taken over all embeddings \(f\) of \(G\) into \(H\).

The wirelength problem [8, 11, 20] of a graph \(G\) into \(H\) is to find an embedding of \(G\) into \(H\) that induces the wirelength \(WL(G, H)\). The following problem has been considered in the literature [21], and is \(NP\)-complete [23].

**Edge Isoperimetric Problem:** Let \(G = (V, E)\) be a graph and \(A \subseteq V\). Denote

\[
I_G(A) = \{(uv) \in E \mid u, v \in A\},
\]

\[
\theta_G(A) = \{(uv) \in E \mid u \in A, v \notin A\}
\]

and

\[
I_G(m) = \max_{A \subseteq V, |A| = m} |I_G(A)|,
\]

\[
\theta_G(m) = \min_{A \subseteq V, |A| = m} |\theta_G(A)|.
\]

For a given \(m\), where \(m = 1, 2, \ldots, n\), we consider the problem of finding a subset \(A\) of vertices of \(G\) such that \(|A| = m\) and \(\theta_G(A) = \theta_G(m)\). Such subsets are called optimal [21, 22]. Moreover, for a regular graph \(G\), \(I_G\) and \(\theta_G\) are equivalent in the sense that a solution for one also becomes a solution for the other [21]. The problem of finding \(I_G\) is called maximum subgraph problem [23].

The following results are powerful tools to find wirelength of an embedding using edge isoperimetric problem.

**Lemma 1.** (Modified Congestion Lemma) [5] Let \(f\) be an embedding of a graph \(G\) into \(H\) with same order. Let \(S\) be an edge cut of \(H\) such that the removal of edges of \(S\) separates \(H\) into exactly 2 connected components \(H_1\) and \(H_2\) and let \(G_1 = f^{-1}(H_1)\) and \(G_2 = f^{-1}(H_2)\). Furthermore, \(S\) satisfies the following conditions:

(i) For every edge \((a, b) \in G_i\), \(i = 1, 2\), \(P_f(a, b)\) has no edges in \(S\).
(ii) For every edge \((a, b) \in G\) with \(a \in G_1\) and \(b \in G_2\), \(P_f(a, b)\) has exactly one edge in \(S\).
(iii) \(G_1\) and \(G_2\) are optimal sets.

Then \(EC_f(S)\) is minimum and

\[
EC_f(S) = \sum_{v \in V(G_1)} deg_G(v) - 2|E(G_1)|
\]

\[
= \sum_{v \in V(G_2)} deg_G(v) - 2|E(G_2)|
\]

**Remark 1.** When the guest graph \(G\) is regular, it is enough to check whether \(G_1\) is an optimal set in condition (iii) of Modified Congestion Lemma [8].

**Lemma 2.** (Partition Lemma) [5] Let \(f : G \to H\) be an embedding. Let \(\{S_1, S_2, \ldots, S_p\}\) be a partition of \(E(H)\) such that each \(S_i\) is an edge cut of \(H\). Then

\[
WL_f(G, H) = \sum_{i=1}^p EC_f(S_i).
\]

**Definition 4.** [10, 14] The undirected circulant graph \(G(n; \pm S)\), \(S \subseteq \{1, 2, \ldots, j\}\), \(1 \leq j \leq [n/2]\), is a graph with the vertex set \(V = \{0, 1, \ldots, n-1\}\) and the edge set \(E = \{(i, k) : |k-i| \equiv s(\mod n), s \in S\}\).

The circulant graph shown in Figure 2 is \(G(8; \pm \{1, 3, 4\})\). It is clear that \(G(n; \pm 1)\) is the undirected cycle \(C_n\) and \(G(n; \pm \{1, 2, \ldots, [n/2]\})\) is the complete graph \(K_n\). The cycle \(G(n; \pm 1)\) contains in \(G(n; \pm \{1, 2, \ldots, j\})\), \(1 \leq j \leq [n/2]\) is sometimes referred to as the outer cycle \(C\) of \(G\).

**Definition 5.** [24] The basic skeleton of a hypertree \(HT(r)\) is a complete binary tree \(T_r\), that is, \(T_r\) is a spanning subgraph of \(HT(r)\), where \(r\) is the level of the tree. Its vertices are labeled as follows: The root node has label 1 and is said to be at level 1. The labels of the left (resp. right) children of a vertex are formed by appending 0 (resp. 1) to the label of the parent vertex, see Fig. 3(a). In the corresponding decimal labelling of the hypertree, the children of the vertex \(x\) are
labeled with $2x$ and $2x+1$. Additional edges in a hypertree are horizontal, where two vertices in the same level $i$, $1 \leq i \leq r$, are joined by an edge if their label difference is $2^{i-2}$, see Fig. 3(b). We denote the $r$-level hypertree with $HT(r)$, $r \geq 2$. The rooted hypertree $RHT(r)$ is obtained from the hypertree $HT(r)$ by attaching to its root a pendant edge. The new vertex is called the root of $RHT(r)$ and is considered to be at level 0.

**Definition 6.** [6] A graph $CH_t$ is a chord graph on $n = 2^t$ nodes with the following vertex and edge sets: $V(CH_t) = \{v_0,v_1,v_2,\ldots,v_{2^t-1}\}$ and $e = (u,v) \in E(CH_t)$ if and only if $i + 2^k \equiv_{\text{mod}_{2^t}} j$ or $j + 2^k \equiv_{\text{mod}_{2^t}} i$, for some $k \in \{0,1,\ldots,t-1\}$: we say that the length of $e$ is $2^k$.

**Remark 2.** The chord graph $CH_t$ is isomorphic to the undirected circulant graph $G(2^t; \{0,2,\ldots,2^t-1\})$, $t \geq 2$.

**Theorem 1.** [9] A set of $k$ consecutive vertices of $G(v; \pm 1)$, $1 \leq k \leq n$, induces a maximum subgraph of $G(v; \pm S)$, where $S = \{1,2,\ldots,j\}$, $1 \leq j < [n/2]$, $n \geq 3$.

**III. WIRELENGTH OF AN EMBEDDING**

Even though there are numerous results and discussions on the wirelength problem, most of them deal with only approximate results and the level of lower bounds [11]. In this section, we produce exact wirelength of an embedding circulant networks into hypertrees and hyper trees into chord graphs.

**A. Circulant networks into hypertrees**

**Wirelength Algorithm A**

**Input:** The circulant network $G(2^r-2; \pm \{1,2,\ldots,2^r-3\})$ and the $r$-dimensional hypertree $HT(r)$, $r \geq 3$.

**Algorithm:** Label the consecutive vertices of $G(2^r-2; \pm 1)$ in $G(2^r-2; \pm \{1,2,\ldots,2^r-3\})$ as $0,1,2,\ldots,2^r-3$ in the clockwise sense and label the vertices of $HT(r)$ as follows: Since $T_r$ is a spanning tree of $HT(r)$, label the root vertex of $T_r$ as $x$ and the remaining vertices using inorder labeling [25] such that for any horizontal edge $e = (u,v) \in HT(r)$, the sum of the labels of $u$ and $v$ is equal to $2^r-3$. See Figure 3. Let $f(x) = x$ for all $x \in V(G)$ and for $(a,b) \in E(G)$, let $P_f(a,b)$ be a shortest path between $f(a)$ and $f(b)$ in $HT(r)$.

**Output:** An embedding $f$ of $G(2^r-2; \pm \{1,2,\ldots,2^r-3\})$ into $HT(r)$ with optimal wirelength.

**Theorem 2.** Let $G$ be the circulant network $G(2^r-2; \pm \{1,2,\ldots,2^r-3\})$ and $H$ be the $r$-dimensional hypertree $HT(r)$, $r \geq 3$. Then the wirelength of $G$ into $H$ is given by

$$WL(G,H) = \left(\frac{1}{3} 4^r - 2^r - 6\right)r - \frac{61}{36} 4^r + 25 \times 2^{-r-1}.$$  

**Proof.** Label the vertices of $G$ and $H$ using Wirelength Algorithm. We assume that the labels represent the vertices to which they are assigned. First we claim that, for any embedding $f : G \rightarrow H$, congestion on edge $xy$ is 0, where $x$ is the label of the root vertex. Since no vertex in $G$ is mapped onto $x$ and for any two vertices $a$ and $b$ in $H$, no shortest path between $a$ and $b$ passes through $x$, the congestion on edge $xy$ is 0.

Let $A^1 = \{(2^{r-2} - 1, x), (2^{r-1} + 2^{r-2} - 2, x)\}$. By our claim $EC_f(A^1) = 0$. Let $A^2 = \{((x,2^{r-2} - 1), (k,2^{r-3} - k) : 0 \leq k \leq 2^{r-2} - 1\}$. For $1 \leq i \leq r-2$, $1 \leq j \leq 2^{r-(i+1)}$ and $j$ is odd, let $B^j_i = \{(2^{r-1}\{2j - 1, 1, 2^{r-1}\{2j - 1, 2j - 1\}, 2^{r-3} - (2^{r-1}\{2j - 1, 1\})\}$. For $1 \leq i \leq r-2$, $1 \leq j \leq 2^{r-(i+1)}$ and $j$ is even, let $B^j_i = \{(2^{r-1}\{2j - 1, 2^{r-1}\{2j - 2, 1\}, 2^{r-1} - (2^{r-1}\{2j - 2, 1\})\}$. Let $A^2 \cup \left\{B^1_i : 1 \leq i \leq r-2, 1 \leq j \leq 2^{r-(i+1)} \right\}$ \{(2^{r-1} + 2^{r-2} - 2, x)\} is a partition of $E(HT(r))$. See Figure 3. Now $E(HT(r)) \{A^2\}$ has two components $H_{21}$ and $H_{22}$, where $V(H_{21}) = \{0,1,2,\ldots,2^{r-1}-2\}$. Let $G_{21} = f^{-1}(H_{21})$ and $G_{22} = f^{-1}(H_{22})$. Then $G_{21}$ is an optimal set and $A^2$ satisfies conditions (i), (ii) and (iii) of Modified Congestion Lemma. Therefore $EC_f(A^2)$ is minimum. For each $i,j$, $1 \leq i \leq r-2$, $1 \leq j \leq 2^{r-(i+1)}$, $E(HT(r)) \{B^j_i\}$ has two components $H_{1j}^1$ and $H_{2j}^1$, where $V(H_{1j}^1) = \{(j-1)2^j, (j-1)2^j+1, \ldots, j2^j, 2^j - 3 - (j-1)2^j, 2^j - 3 - (j-1)2^j+1, \ldots, 2^j - 3 - (2^j-2)\}$. Let $G_{1j}^1 = f^{-1}(H_{1j}^1)$ and $G_{2j}^1 = f^{-1}(H_{2j}^1)$. The subgraph $G_{1j}^1$ is optimal, since it induces a complete graph on $2^{r-1} - 2$ vertices and each $B^j_i$ satisfies conditions (i), (ii) and (iii) of Modified Congestion Lemma. Therefore $E(f(B^j_i))$ is minimum. The Partition Lemma implies that the wirelength is minimum. By Modified Congestion Lemma,

(i) $EC_f(A^2) = 2(2^{r-1} - 1)(2^{r-1} - 1) - \frac{1}{2}(2^{r-1} - 3)(2^{r-1} - 2)$

(ii) $EC_f(B^j_i) = 2(2^{r-1} - 1)(2^{r-1} - 2) - \frac{1}{2}(2^{r-1} - 3)(2^{r-1} - 2)$

for $1 \leq i \leq r-2$, $1 \leq j \leq 2^{r-(i+1)}$.
Then, by Partition Lemma,

\[ WL(G, H) = \begin{cases} 
(2^{r-1} - 3)(2^{r-1} - 2) \\
+ \sum_{i=1}^{r-2} \sum_{r+1}^{2r-2} (2^{r+1} - 2)(2^{r} - 2^{r+1} - 3) \\
= \left( \frac{1}{3} 4^r - 2^r - 6 \right)r - \frac{61}{36} 4^r + 25 \times 2^{r-1}. 
\end{cases} \]

B. Hypertrees into chord graphs

The chord graph is a powerful topology in the area of peer-to-peer networks. Thus it is interesting to study the embedding problems on chord graphs. Further, the chord graph \( CH_r \) is isomorphic to the undirected circulant graph \( G(2^r; \pm\{2^0, 2^1, \ldots, 2^{r-1}\}) \), \( r \geq 2 \). Moreover, hypercubes and generalized hypercubes are subgraphs of chord graphs.

We now prove that hypertree is a subgraph of chord graph.\( \) thereby proving that wirelength of embedding hypertree into chord graph is the number of edges in the hypertree.

Wirelength Algorithm B

**Input**: The rooted hypertree \( RHT(r) \) and the chord graph \( CH_r, r \geq 3. \)

**Algorithm**: Label the vertices of \( RHT(r) \) as follows:

Removal of the horizontal edges in rooted hypertree \( RHT(r) \) leaves a rooted complete binary tree \( RT_r \). Label the vertices in level 0 and level 1 as 0 and 1 respectively. For \( 1 \leq i \leq r \), the children of the vertex \( x \) in the level \( i \) are labeled as \( 2^i - 1 + x \) and \( 2^i + x \). Label the consecutive vertices of \( G(2^r; \pm\{2^0, 2^1, \ldots, 2^{r-1}\}) \), \( 0 \leq i \leq 2^r - 1 \), in the clockwise sense. See Figure 4. Let \( f(x) = x \) for all \( x \in V(RHT(r)) \) and for \( (a, b) \in E(RHT(r)) \), let \( P_f(a, b) \) be a shortest path between \( f(a) \) and \( f(b) \) in \( CH_r \).

**Output**: An embedding \( f \) of \( RHT(r) \) into \( CH_r \) with wirelength \( 3 \times 2^{r-1} - 4 \).

**Proof of correctness**: Label the vertices of \( RHT(r) \) and \( CH_r \) using Wirelength Algorithm B. We assume that the labels represent the vertices to which they are assigned.

Let \( u \) be any vertex in \( RHT(r) \) with label \( x \). We define a function \( g \) from \( V(RHT(r)) \) to \( V(CH_r) \) as follows:

\[ g(x) = x. \]

The function \( g \) is obviously bijective. Let \( u \) and \( v \) be two distinct vertices in \( RHT(r) \) with label \( x \) and \( y \) respectively. It follows that \( g(x) \) and \( g(y) \) are the labels of two distinct vertices in \( CH_r \) given as follows:

\[ g(x) = x, \quad g(y) = y. \]

Let the labels \( x \) and \( y \) be adjacent in \( RHT(r) \). Then \( |y - x| = 2^j \) for some \( 0 \leq j \leq r - 1 \). This implies \( g(x) \) and \( g(y) \) are adjacent in \( CH_r \). Hence, the rooted hypertree \( RHT(r) \) is a subgraph of \( CH_r \). In other words, \( dil(RHT(r), CH_r) = 1 \).

The following theorem is a consequence of Wirelength Algorithm B.

**Theorem 3**: Let \( G \) be the rooted hypertree \( RHT(r) \) and \( H \) be the chord graph \( CH_r, r \geq 3 \). Then the wirelength of \( G \) into \( H \) is given by

\[ WL(G, H) = |E(G)| = 3 \times 2^{r-1} - 4. \]

IV. Time Complexity

In computer science, the time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the size of the input to the problem. An algorithm is said to take linear time, or \( O(n) \) time, if its time complexity is \( O(n) \). Informally, this means that for large enough input sizes the running time increases linearly with the size of the input [26]. Linear time is often viewed as a desirable attribute for an algorithm. Much research has been invested into creating algorithms exhibiting (nearly) linear time or better. This research includes both software and hardware methods. In the case of hardware, some algorithms which, mathematically speaking, can never achieve linear time with standard computation models are able to run in linear time. There are several hardware technologies which exploit parallelism to provide this. An example is content-addressable memory. This concept of linear time is used in string matching algorithms such as the Boyer Moore Algorithm and Ukkonen Algorithm [26].

In this section, we compute the time complexity of finding the exact wirelength of embedding circulant networks into hypertrees using Wirelength Algorithm A. The algorithm is formally presented as follows.

**Time Complexity**

**Input**: The circulant network \( G(2^r - 2; \pm\{1, 2, \ldots, 2^{r-1} - 3\}) \) and the \( r \)-dimensional hypertree \( HT(r), r \geq 3 \).

**Algorithm**: Wirelength Algorithm A

**Output**: The time taken to compute the minimum wirelength of embedding \( G(2^r - 2; \pm\{1, 2, \ldots, 2^{r-1} - 3\}) \) and the \( r \)-dimensional hypertree \( HT(r), r \geq 3 \) is \( O(n^2) \).

**Method**: We know that \( G \) contains \( n = 2^r - 2 \) vertices. For assigning the labels of \( n \) vertices, we spend \( n \) time units. By Wirelength Algorithm A, we have \( 2^{2r-6} + 2^{2r-3} + 1 \) edge cuts. For each edge cut \( C_i, 1 \leq i \leq 2^{2r-6} + 2^{2r-3} + 1 \), we need one unit of time and hence we need \( 2^{2r-6} + 2^{2r-3} + 1 \) time units. Again for finding the edge congestion on \( C_i, 1 \leq i \leq 2^{2r-6} + 2^{2r-3} + 1 \) we need one unit of time. Further, we need one unit of time for finding the wirelength by using Partition Lemma. Hence, the total time is

\[ n + 2^{2r-6} + 2^{2r-3} + 1 + 2^{2r-6} + 2^{2r-3} + 1 = n + (2^{2r-5} + 2^{2r-2} + 2 + 1) = 2^r(2^{2r-5} + 1 + \frac{1}{4}) < n^2 + n \]

\[ = O(n^2) \]
Hence the time taken to compute the exact Wirelength of embedding circulant networks into hypertrees is \( O(n^2) \).

V. CONCLUDING REMARKS

In this paper we compute the wirelength of an embedding circulant graph into hypertrees and hypertrees into chord graphs. Further, we compute the time complexity of finding the exact wirelength of embedding circulant networks into hypertrees and vice-versa. Finding the other parameter such as congestion of embedding circulant networks into hypertrees and vice-versa is under investigation.

REFERENCES