Distributed Algorithms of Finding the Unique Minimum Distance Dominating Set in Directed Split-Stars

Fu–Hsing Wang1 Jou–Ming Chang2 Yue–Li Wang1,

1Department of Information Management, National Taiwan University of Science and Technology
2Department of Information Management, National Taipei College of Business

Abstract

A distance-$k$ dominating set $S$ of a directed graph $D$ is a set of vertices such that for every vertex $v$ of $D$, there is a vertex $u \in S$ at distance at most $k$ from it. Minimum distance-$k$ dominating set is especially important in communication networks for distributed data structure and for server placement. In this paper, we shall present simple distributed algorithms for finding the unique minimum distance-$k$ dominating set for $k = 1, 2$ in a directed split-star, which has recently been developed as a new model of the interconnection network for parallel and distributed computing systems.

1 Introduction

Star graphs provide attractive interconnection scheme for parallel computers and distributed networks, hence characterizations of this architecture has been widely investigated [1, 2, 6]. In [5], Cheng et al. gave a variant architecture of star graphs which is namely split-stars. Later on, Cheng and Lipman [4] proposed an assignment of directions to the edges of split-stars as a new class of digraph called the directed split-stars. They also showed that the resulting digraphs are not only strongly connected, but, in fact, they have maximal arc-fault tolerance and a small diameter. Recently, Wang et al. [11] showed that the directed split-stars have the unique minimum distance-$k$ dominating set for $k = 1, 2$. In this paper, we shall present distributed algorithms for finding such sets.

Let $D = (V, A)$ be a directed graph (digraph) with vertex set $V$ and arc set $A$, where $A \subseteq V \times V$. An arc $<u, v>$ is said to be directed from $u$ to $v$, in which case we say that $u$ dominates $v$, and $v$ is dominated by $u$. Let $k$ be a positive integer. A set $S$ of vertices in a digraph $D$ is called a distance-$k$ dominating set if for every vertex $v$ of $D$, there is a vertex $u \in S$ at distance at most $k$ from it. A minimum distance-$k$ dominating set is a distance-$k$ dominating set with the minimum cardinality.
Researches on domination and related topics have attracted graph theorists due to the strongly practical applications and theoretical interesting. Efficient construction for distance-\(k\) dominating sets can be applied in the context of distributed data structure [10], where it is proposed that a distance-\(k\) dominating set can be selected for locating copies of a distributed directory. Likewise, such a set is useful for efficient selection of network centers for server placement, where it is desired to ensure that each node in the network is sufficiently close to some server [3]. A thorough treatment of domination in graphs and digraphs can be found in [7, 8].

The remaining part of this paper is organized as follows. In Section 2, we give the definition of directed split-stars and introduce some basic terminologies and notations. In Section 3, we present the distributed algorithms for finding the unique minimum distance-\(k\) dominating set for \(k = 1, 2\), and show the correctness of the proposed algorithms. Finally, a concluding remark is given in the last section.

## 2 Preliminaries

For a digraph \(D = (V, A)\) and \(u, v \in V\), the distance \(d(u, v)\) is the number of arcs along a shortest path from \(u\) to \(v\), and \(d(u, v) = \infty\) if there is no path from \(u\) to \(v\) in \(D\). Let \(k\) be a positive integer. The distance-\(k\) outset of a vertex \(u\) in \(D\) is the set \(O_k(u) = \{v \in V \mid d(u, v) = k\}\), while the distance-\(k\) inset is \(I_k(u) = \{v \in V \mid d(v, u) = k\}\). For simplicity, we write \(O(u)\) and \(I(u)\) for \(k = 1\). Let \(od(u)\) and \(id(u)\) denote the outdegree and the indegree, respectively, of a vertex \(u \in V\), i.e., \(od(u) = |O(u)|\) and \(id(u) = |I(u)|\). For terms of digraphs not defined here please refer to [9].

![Figure 1: 4-dimensional directed split-star.](image)

The \(n\)-dimensional directed split-star \(S^2_n\) is a digraph whose vertices are in a one-to-one correspondence with \(n!\) permutations \([p_1, p_2, \ldots, p_n]\) of the set \(N = \{1, 2, \ldots, n\}\), and two vertices \(u, v\) of \(S^2_n\) are connected by an arc \(<u, v>\) if and only if the permutation of \(v\) can be obtained from \(u\) by either a 2-exchange or a 3-rotation. Let \(u = [p_1, p_2, \ldots, p_n]\). A 2-exchange interchanges the first symbol \(p_1\) with the second symbol \(p_2\) whenever \(p_1 > p_2\), in which case \(v = [p_2, p_1, \ldots, p_n]\) is called a 2-exchange neighbor of \(u\). A 3-rotation rotates the symbols in positions 1, 2 and \(i\) counterclockwise for some \(i \in \{3, 4, \ldots, n\}\), and in this case \(v = [p_i, p_1, p_3, \ldots, p_{i-1}, p_2, p_i+1, \ldots, p_n]\) is said to be a 3-rotation neighbor of \(u\). Figure 1 depicts an example of \(S^2_n\) for \(n = 4\).

Let \(u = [p_1, p_2, \ldots, p_n]\). It is easy to see from the definition that the outdegree of \(u\) is \(n - 1\) and the indegree \(n - 2\) if \(p_1 > p_2\);
for otherwise, \( u \) has the outdegree \( n - 2 \) and the indegree \( n - 1 \). To simplify the description of our results, we define the following sets. Let \( N_i = N \setminus \{i\} \) for each \( i = 1, 2 \), and let \( N_{12} = N \setminus \{1, 2\} \), where \( N = \{1, 2, \ldots , n\} \). In [11], Wang et al. showed the following result.

**Theorem 1** Let \( S = \{[p_1, 1, p_3, \ldots , p_n] \mid p_i \in N_1, i \in N_2\} \) and \( T = \{[2, 1, p_3, \ldots , p_n] \mid p_3, \ldots , p_n \in N_{12}\} \). Then, \( S \) and \( T \) are the unique minimum distance-k dominating set of \( \overrightarrow{S}^2_n \) for \( k = 1, 2 \), respectively.

A distributed network can be viewed as a system consisting two types of components: processors and interconnections between processors. Usually, such a system is modeled as a digraph where each vertex representing a processor, and a directed arc between processors representing message channel for communications. A **distributed algorithm** is defined to be a collection of local algorithms from the same copy that one for each processor. Every processor independently executes its local algorithm and cooperates with the other processors to achieve a certain objective. To communicate among themselves, the processors share information through transmitting messages. That is, the processors can communicate only by sending and receiving messages over the communication links. In this paper, we assume that individual processor operates at a different speed with respect to other processors and the timing model used in the communications network is asynchronous, i.e., the transmitting message is placed in a queue waiting for the receiver to accept it and the sending processor can proceed immediately. We also assume that each vertex (processor) \( u \) knows the indegree \( id(u) \) and the outdegree \( od(u) \), while the identities of all vertices in the graph are unknown. Since the graph dimension \( n \) and the larger value of \( id(u) \) and \( od(u) \) differ by 1, each processor knows the network size. Note that each processor has a local state and all variables are locally maintained by each processor.

### 3 The distributed algorithms

In the rest of this paper we assume that \( S = \{[p_1, 1, p_3, \ldots , p_n] \mid p_i \in N_1, i \in N_2\} \) and \( T = \{[2, 1, p_3, \ldots , p_n] \mid p_3, \ldots , p_n \in N_{12}\} \) are the unique minimum distance-k dominating set of \( \overrightarrow{S}^2_n \) for \( k = 1, 2 \), respectively. Before introducing our algorithm for computing the distance-1 dominating set in a directed split-star, we first show a property related to the domination which will be used for proving the correctness of the algorithm.

**Lemma 2** Let \( x \) be a vertex in \( \overrightarrow{S}^2_n \). Then \( x \in S \) if and only if \( od(y) < id(y) \) for every \( y \in I(x) \).

**Proof.** Suppose \( x = [p_1, 1, p_3, \ldots , p_n] \) and \( y \in I(x) \). It is clear that \( x \) is a 3-rotation neighbor of \( y \). Let \( y = [1, p_i, p_3, \ldots , p_{i-1}, p_1, p_{i+1}, \ldots , p_n] \) for some \( i \in N_{12} \). Since the symbol 1 is on the first position of \( y \), \( od(y) = n-2 < id(y) = n-1 \).

Conversely, we assume for the purpose of contraction that \( x \notin S \). We will obtain the contracting result that there is a vertex \( y \in I(x) \) such that \( id(y) < od(y) \). Consider the following two cases.

**Case 1:** the symbol 1 is on the first position of \( x \). Let \( x = [1, p_2, p_3, \ldots , p_n] \) and \( y = [p_2, 1, p_3, \ldots , p_n] \). Clearly, \( x \) is a 2-exchange neighbor of \( y \). Since the symbol 1 is on the second position of \( y \), \( id(y) = n-2 < od(y) = n-1 \).

**Case 2:** the symbol 1 is on the \( i \)th position of \( x \) where \( 3 \leq i \leq n \). Let
$x = [p_1, p_2, p_3, \ldots, p_{i-1}, 1, p_i+1, \ldots, p_n]$ and $y = [p_2, 1, p_3, \ldots, p_{i-1}, p_1, p_i+1, \ldots, p_n]$. Clearly, $x$ is a 3-rotation neighbor of $y$. Since the symbol 1 is on the second position of $y$, $\text{id}(y) < \text{od}(y)$. \hfill \Box

According to Lemma 2, we design an algorithm for computing the distance-1 dominating set with two stages. In the first stage, each processor sets up the message which is 0 or 1 and sends the message to its neighbors. In the second stage, the received messages are summed and compared with the indegree in each processor for determining the membership of the dominating set. Below we present the details of the algorithm and let $x$ be a processor for running the algorithm.

**Algorithm** Distance-1 Domination

Stage 1:
Set up the transmitting message to be 1 if $x$’s outdegree is less than its indegree, and to be 0 otherwise;
Send the set up message to each neighbor $y \in O(x)$;

Stage 2:
Receive the incoming messages and count the number of 1’s until all messages are received;
If the sum accumulated by $x$ is equal to $x$’s indegree then $x \in S$, otherwise $x \notin S$;

**Theorem 3** Algorithm Distance-1 Domination correctly determines the unique minimum distance-1 dominating set of $S_n^2$.

**Proof.** From the algorithm, it is easy to see that the sum accumulated by $x$ indicates the number of vertices directed to $x$ with the outdegree less than the indegree. By Lemma 2, $x$ is a member of the unique minimum distance-1 dominating set if and only if each vertex directed to $x$ has the outdegree less than the indegree, and therefore it can be determined by the last statement of Stage 2. \hfill \Box

We now give the complexity analysis as follows. Since each message is transferred only once in the first stage of the algorithm, the communication complexity is evidently proportional to the number of directed arcs in the digraph.

For computing the distance-2 dominating set of $S_n^2$, we design an algorithm having three stages where the first two stages are the same with the previous algorithm, excepting that the sum accumulated by $x$ is sent out to $x$’s neighbors again. Then, in the third stage the secondly received messages are compared with a certain condition for determining the membership of the distance-2 dominating set. Since in the algorithm each processor receives message twice from every neighbor and these incoming messages must be discriminated for controlling the sequence of stages, we may assume that a tag is provided for every message in order to indicate the different type of messages.

**Algorithm** Distance-2 Domination

Stage 1-2.
Each processor $x$ performs the same two stages with Algorithm Distance-1 Domination, and send the sum as a transmitting message to each neighbor $y \in O(x)$;

Stage 3.
Receive the incoming messages and count them until all messages are received;
If $x \in S$ and the total counted sum equals to $\frac{(n-2)(n-3)}{2}$ then $x \in T$, otherwise $x \notin T$;
To show the correctness of the algorithm, we need the following lemma.

**Lemma 4** Let $x = [p_1, p_2, p_3, \ldots, p_n]$ and $x \in S$. Then the following statements are true:

(i) If $y \in I_2(x)$ and $y = [y_1, y_2, \ldots, y_n]$, then $y_i \neq \{1, p_1\}$.

(ii) If $Y = \{y \in I_3(x) \mid od(y) < id(y)\}$, then $|Y| = [1 + 2 + \cdots + (n - 2)] - (n - p_1)$.

**Proof.** Since $p_1 > p_2 = 1$, $x$ cannot be a 2-exchange neighbor of any vertex. Suppose $y \in I_2(x)$ and let $w$ be a vertex in $I(x)$ such that $y \in I(w)$. Since $x$ is a 3-rotation neighbor of $w$, we have

$$w = [p_2 = 1, p_3, \ldots, p_{i-1}, p_i, p_{i+1}, \ldots, p_n]$$

for some $i \in \{3, 4, \ldots, n\}$. (1)

Regardless of the fact that $w$ is either a 2-exchange or a 3-rotation neighbor of $y$, it is easy to see that $p_i$ must be put in the first position of $y$. Since $p_i \in N_1$ and $p_i \neq p_1$ for $i = 3, \ldots, n$, this shows that statement (i) holds.

To show the statement (ii), we first define $Y_w = \{y \in I(w) \mid od(y) < id(y)\}$ for each vertex $w \in I(x)$. We now consider a vertex $y \in Y_w$. Since $od(y) < id(y)$, it implies that $y$ is a 2-exchange neighbor of some vertex. Thus, if $y_1$ and $y_2$ are the symbols on the first position and the second position, respectively, of $y$, then $y_1 < y_2$. In particular, $w$ is a 3-rotation neighbor of $y$ in this case. According to (1), we can represent

$$y = [p_i, p_j, p_3, \ldots, p_{j-1}, p_2 = 1, p_{j+1}, \ldots, p_n]$$

where $j \in N_2 \{i\}$.

By the fact $p_i < p_j$, we have $|Y_w| = n - p_i$. Indeed, the above argument shows that all vertices in $Y_w$ have the same symbol, say $p_i$, on the first position. Thus $Y_w \cap Y_{w'} = \emptyset$ for any two vertices $w, w' \in I(x)$. Note that

$$Y = \{y \in I_2(x) \mid od(y) < id(y)\} = \bigcup_{w \in I(x)} \{y \in I(w) \mid od(y) < id(y)\} = \bigcup_{w \in I(x)} Y_w$$

Therefore, we have the following consequence.

$$|Y| = \sum_{w \in I(x)} |Y_w| = \sum_{p_i \in N_1 \{p_1\}} n - p_i = \left[\sum_{k=2}^{n} (n - k)\right] - (n - p_1)$$

This completes the proof. □

The following corollary directly follows from the fact $T \subset S$ and the statement (ii) of Lemma 4.

**Corollary 5** Suppose $x \in S$ and let $Y = \{y \in I_2(x) \mid od(y) < id(y)\}$. Then $x \in T$ if and only if $|Y| = 1 + 2 + \cdots + (n - 3)$.

**Theorem 6** Algorithm DISTANCE-2 DOMINATION correctly determines the unique minimum distance-2 dominating set of $S_n^3$.

**Proof.** Since $T \subset S$, the vertices of $T$ are filtered through $S$ by the algorithm. Let $x$ be a vertex in $S$. Since the total sum accumulated by $x$ indicates the number of vertices with the distance 2 from $x$ and satisfying the property that the outdegree is less than the indegree, the correctness of the algorithm directly follows from Corollary 5 and the last statement of Stage 3. □

## 4 Concluding Remark

Wang et al. [11] showed that a directed split-star has the unique minimum
distance-$k$ dominating set for $k = 1, 2$. In this paper, we propose distributed algorithms for finding such dominating sets. However, the problem of finding a minimum distance-$k$ dominating set for $k \geq 3$ on directed split-stars is still unsolved.

References


