論 文

An Algorithm for Construction of Distributed Breadth-First Search Tree Using New Threshold Values

In Sop SONG*, Jae Ho SHIN** *Regular Members*

새로운 임계값을 이용한 분산 너비우선탐색 트리 (Distributed Breadth-First Search Tree)의 구성에 관한 알고리즘

正會員 宋 仁 燮* 正會員 申 宰 浩**

ABSTRACT In construction of breadth-frist tree, the communication complexity can be reduced by efficent synchronization schemes based on several threshold values. We determine several new threshold values by considering the graph density represented as $\log_n m$, where n and m are the number of nodes and links, respectively. When these threshold values are used in the synchroization method for constructing distributed breadth-first search tree, we can obtain a more efficient algorithm in sparse graphs, and also, this alorithm has the same performance for communication complexity in dense graphs

要 約 분산 너비우선탐색 트리 구성에서 통신복잡성(Communication Complexty)은 몇가지 임계값에 기초한 효과적인 통신동기방식에 의해서 개선될 수 있다.

본 논문에서는 분산 그래프의 밀도함수로써 임계값을 설정하고 그 임계값을 이용하여 통신동기방식에 의거한 분산 너비우선탐색 트리 구성 알고리즘을 제안하였다. 제안된 알고리즘은 밀도가 낮은 그래프에서 기존의 알고리즘보다 통신복잡성 면에서 수학적 분석을 통해 개선됨을 입증하였으며, 밀도가 높은 그래프에서는 현재와 동일함을 입증함으로써 본 논문에서 제안한 알고리즘이 종래의 알고리즘들보다 통신복잡성에서 가장 효율적임을 보였다.

I. Introduction

Algorithms for finding distributed breadth-first search tree(DBFST) can be used as a key component for determining topological informations on graphs, such as diameters, centers, and all pairs shortest paths on graphs. For constructing DBFST in a synchronous communication model, synchronization is required to explore next level for connecting with shortest path between parent nodes and child nodes and this synchronization incurs much communication overheads for constructing DBFST. Many works on algorithms for constructing DBFST on graphs have concentrated on reducing the cost of synchronization ⟨1, 3, 5⟩, and several synchronization scheme based on clustering have been suggested as an improvement techniques ⟨1, 3, 5⟩.

Basically, synchronization methods for construcuting DBFST can be categorized into three approaches : α-synchronization, β-synch-

*生產技術硏究院 電氣電子應用室
Korea Academy of Industrial Technology(KAITECH)
**東國大學校 電子工學科
Dept. of Electronic Eng., Dongguk University
論文番號 : 91-45(接受1991. 2. 6)

ronization and cluster-based synchronization 〈3〉. In 〈4〉, an algorithm usning α-synchronization is proposed whose communication complexity is bounded by $O(mn)$. In 〈6〉, an algorithm using β-synchronization is proposed whose communication complexity is bounded by $O(n^2)$. The cluster-based synchronization is more improvement in the technique of optimizing the cost of synchronization than α-synchronization and β-synchronization and this synchronization method is a merged form of α-synchronization and β-synchronization. Also, this synchronization method is based on several threshold values and these threshold values affect on the cost of synchronization as an important factor.

In the algorithm of 〈5〉, each cluster consistes of $\frac{n}{m}$ levels and the α-synchronization is used within each cluster and the β-synchronization is used between clusters. This algorithm leads to a communication complexity of $O(n\sqrt{m})$.

In the algorithm of 〈1〉, each cluster is constrained sophisticatedly by several threshold values represented as functions of the number of nodes. This algorithm is similar to that of 〈4〉 in that it is based on the β-synchronization, synchronization only among successive clusters of nodes. The difference is that the number of levels processed in each cluster is a function of the value of n.

Another difference is that the synchronization among clusters are performed more efficiently by considering the number of synchronization nodes and providing an efficient synchronization mechanism for nodes with many incident links. The synchronization mechanism of 〈1〉 is based on several threshold values represented as functions of the number of nodes in a graph. The algorithm using these threshold values requires the communication complexity of $O(m+n^{1.6})$, where n and m are the number of nodes and links, respectively 〈1〉.

In this paper, we propose and improvement algorithm which uses the same synchronization scheme in 〈1〉, but uses new threshold values. These new threshold values are represented as functions of m and n, unlike to 〈1〉, since the value of m affects on the message complexity as an important factor. These threshold values are determined in a piecewise manner according to the value of $\log_n m$, called graph density, and the algorithm using these threshold values has more advantage than other existing algorithms in the communication complexity, and produces a more efficient solution in sparse graphs.

Ⅱ. Representation of the communication complexity for constructing DBFST

Awerbuch and Gallager had proposed an algorithm 〈1〉 by using the threshold values represented as functions of only n, where the value of n can be found within $O(m)$ messages. Also, they suggested a synchronization method based on clustering by using these threshold values. And, this synchronization method is based on global broadcasts and local broadcasts, where global broadcasts and local broadcasts are used for synchronization among th clusters and the nodes in each cluster, respectively. Each synchronization by the global broadcasts is referred to iteration. Trade-offs exist between the overheads due to global broadcasts and local broadcasts. And the optimization in these trade-offs is important for redcuing the communication complexity.

Several threshold values are used to control the trade offs. And the optimization for reducing the communication overheads is tailed to set up the threshold values appropriately. In 〈1〉, improtant factors influencing the trade-offs can be described as follows :

- the overheads for processing the nodes with many incident links.
- the number of levels which each cluster can explore.
- the overheads due to interferences which can occur between other clusters when cluster extends its local breadth first search tree.

The several threshold values which control the above mentioned factors can be summarized as foolows in 〈1〉.

- x=the threshold used to freeze the nodes which has incident deges more than n^x(x is 0.4 in 〈1〉).
- y=the threshold used to limit the number of synchronization nodes periteration and to control the overheads due to interferences between each cluster(y is 0.2 in 〈1〉).
- z=the threshold used to limit the number of levels which cna be extended nominally per iteration(z is 0.2 in 〈1〉).
- $f(\cdot)z$=the threshold used to represent the number of levels to be extended when the number of local trees which can be explored furthermore is less than n after exploring the attempted depth n^z($f(\cdot)$ is 0.4 in 〈1〉).

An increase of the above mentioned threshold values makes the overheads due to the local synchronization much larger and optimization is needed by setting these threshold values appropriately. The communication complexity is expressed as summation of the overheads due to global synchronization and local synchronization, both of which are expressed by the threshold values in 〈1〉. In this section, we describe the communication overheads due to global synchronization and local synchronization by using the threshold values.

Firstly, we describe the communication overheads due to global synchronization. Global synchronization is performed whenever one of the three following conditions occurs :

- After the number of levels explored at an iteration reaches the atempted depth n^z, the number of local trees which can be explored furthermore is counted. Then, if the number of these local trees is greater than n^y, a new iteration is performed. Otherwise, after the local trees are explored to $n^{f(\cdot)z}$ levels furthernore, a new iteration is performed(the number of iterations due to this case is denoted by $c_{g,1}$).
- The tree is frozen at some smaller depth by seizing a node of degree greater than n^x(the number of iterations due to this case is denoted by$c_{g,2}$).

Here, $C_{g,1}$ is bounded by $O(n/\min(n^{f(\cdot)z}, n^{y\cdot z}))$ since one iteration explores $O(\min(n^{f(\cdot)z}), n^{y\cdot z}))$ nodes at least, and $c_{g,2}$ is bounded by $O(n/n^x)$.

Therefore, the communication complexity (denoted by c_g) due to the global broadcast can be expressed as follows.

- The communication complexity due to global broadcasts

$$c_g = n * \text{the number of iterations}$$

$$c_g = n * (c_{g,1} + c_{g,2})$$

$$c = O\left(n\left(\frac{n}{\min(n^{f(\cdot)z}, n^{y\cdot z}} + \frac{n}{n^x}\right)\right) \quad (1)$$

The communication complexity(denoted by c_l) due to local broadcasts can be represented as a summation of the following items :

- The overheads due to the interferences between each cluster(denoted by local synchronization nodes(denoted by c_{l1}).
- The overheads due to the message for initial local broadcast from the local synchronization nodes(denoted by C_{l2}).
- The overheads due to freezing the nodes with incident links more than n^x(denoted by c_{l3})

Here, the overhead c_{l1} con be divided into two case : one case is when the local synchronization node(denoted by y^*) is greater than n^y, and the other case is when y^* is less than n^y. In the first case, the number of level change operations becomes $\min(y^*, n^z)$ ($\leq n^z$). In the latter case, the number of level change operation becomes $\min(y^*, n^{f(\cdot z)})$ ($\leq n^y$). Therefore, $c_{l1}=n^z \times n^x \times n+n^y \times n^z \times n$. The complexity of c_{l1} can be anothre form by considering the total number of edges. One level change of one node makes the node send messages equal to the number of incident deges to the node, and, when every node executes this operation $O(m)$ messages are generate. The number of level change operation of each node is bounded by $O(\text{man}(n^z, n^y))$. Therefore, the complexity of c_{l1} can be represented by $c_{l1}=n^z \times m+n^y \times m$. The overhead c_{l1} for initial local broadcast from the local synchronization nodes, is bounded by $O(m)$, since each node is captured from one among the local synchronization nodes. And after comparing te level of nodes to that node, the overheads for sending to neighbor nodes the messages with the updated level value, are included in [1,6]. The overhead [1,6] is bounded by $O(m)$, since each node of degree more than n^x sends at most one message on each incide[illegible] links as a result of freezing it.

- The communication complexity due to local broadcasts

$$c_l=c_{l1}+_{l2}+c_{l3}$$

$$c_l=O(n^{x\cdot z}m)+O)(n^{x\cdot y}m)+O(m) \qquad (2\text{-}1)$$

$$c_l=O(n^z m)+O(n^y m)+O(m) \qquad (2\text{-}2)$$

Ⅲ. Determination of new threshold values

We assume that the initiation node of the algorithm knows about the values of m and n by a traversal of a given graph, and determines the threshold values, and broadcasts the determined threshold values to other nodes by another graph traversal. Therefore, we assume that each node in a given graph knows every threshold value within $O(m)$ messages before a algortithms for constructing DBFST are executed. The communication complexity is very much dependent on the value of m, and the value of m is reflected on the determination of new threshold values. We derive new threshold values rpresented as functions of m and n, i.e., $\log_n m$, called the graph density.

The new scheme to be described is identical to that of ⟨1⟩ in that it is based on the same synchronization method. The difference is that all threshold values are functions of the graph density. We define parameter ϕ.

$$\phi = \begin{cases} k, & \text{if } 1 \leq k \leq 1.2 \\ 1.6, & \text{if } 1.2 \leq k \leq 1.6 \\ k, & \text{if } 1.6 \leq k \leq 2 \end{cases}$$

where k is $\log_n m (1 \leq k \leq 2)$.

Now the threshold values are represented as functions of ϕ below.

- threshold x

$$x=2-\phi \quad (3)$$

- threshold y

$$y=1-\phi/2 \quad (4)$$

- threshold z

$$z=1-\phi/2 \quad (5)$$

- function f(·)

$$f(\cdot)=x/z \quad (6)$$

Nove we classify the following cases for the value of k and analyze the message complexity in each case.

- Case i) when $1 \leq k \leq 1.2$

With substitution of $\phi=k$ into Eq. (3), (4), (5) and (6), the threshold values of x, y, z and f(·) become $2-k$, $1-k/2$, $1-k/2$ and 2, respectively . By substituting these threshold values into Eq. (1) and (2 2), we obtain the communication complexities due to global broadcasts and local broadcasts become n^k and nm^{k-2}, respectively. Therfore, the communicaton complexity is $O(n\sqrt{m})$ since $nm^{k-2} \geq n^k$ in this case.

- Case ii) when $1.2 \leq k \leq 1.6$

With substitution of $\phi=1.6$into Eq. (3), (4), (5) and (6), the threshold values of x, y, z and f(·) become 0.4, 0.2, 0.2, and 2, respectively. In this case, the derived threshold values are identical to those of 〈1〉. THerefore, the communication complexity can be represented as $O(m+n^{1.6})$. The communication complexity is $O(m)$ since $m \geq n^{1.6}$ in this case.

- Case iii) when $1.6 \leq k \leq 2$

With substitution of $\phi=k$ into Eq. (3),(4), (5) and (6), the threshold values of x, y, z and f(·) become $2-k$, $1-k/2$, $1-k/2$ and 2, respectively. By substituting these threshold values into Eq. (1) and (2 1), we obtain the communication complexities due to global broadcasts and local broadcasts become n and $n^{4-1.5k}$, respectively. In this case, since $n^k \geq n^{4-1.5k}$, therefore, the communication complexity can be represented as

$$O(n^k)(=O(m))$$

By summerizing cases i), ii) and iii), the total communication complexity can be represented as $\min(n\sqrt{m}, m+n^{1.6})$ by using the new threshold values in terms of the graph density. When graph density is less than 1.2, the communication complexity is $O(n\sqrt{m})$ $(\leq O(n^{1.6}))$. And, when graph density is between 1.2 and 1.6 the communication complexity is $O(n^{1.6})$. And, when the graph density is greater than 1.6, the communication complexity is $O(m)$.

Theoretical background to define parameter ϕ

The communication complexities of the algorithms of 〈5〉 and 〈1〉 are given by $n\sqrt{m}$ and $m+n^{1.6}$, respectively. Thus, these communication complexities are identical when graph density is 1.2. And when $1.0 \leq$ graph density $k \leq 1.2$, the algorithm of 〈5〉 is superior to that of 〈1〉. And otherwise, the algorithm of 〈1〉 is superior to that of 〈5〉.

$$n\sqrt{m} \leq m+n^{1.6}$$

When graph density k is less than 1.2, the term of $n^{1.6}$ is dominant to the term of m. Thus,

$$n\sqrt{m} \quad n^{1.6}$$
$$n^{1+k/2} \leqq n^{1.6}$$
$$k \leqq 1.2$$

The proposed algorithm is a unified algorithm that combines the algotrihm of ⟨1⟩ and the algorithm of ⟨5⟩ by using graph density k. While the poposed algorithm behaves like the algorithm of ⟨5⟩, when $1.0 \leqq k \leqq 1.2$. And, it behaves like the algorithm of ⟨1⟩, when $1.2 \leqq k \leqq 2.0$. This is the reason for defining the threshold values differently at $k = 1.2$. Thus when graph density $k \leqq 1.2$, the proposed algorithm is same to the algorithm ⟨5⟩ in terms of communication complexity. And otherwise, the proposed algorithm is same to the algorithm of ⟨1⟩. As results, the proposed algorithm superior to the algorithms of ⟨1⟩ and ⟨5⟩ by taking advantages of two algorithms of ⟨1⟩ and ⟨5⟩ by using the graph density as threshold values.

Ⅳ. Concluding Remarks

Algorithms for finding DBFST can be used as a key component for determintng topological on graphs, and the Awerbuch's algorithm is the most efficient one. In construction of DBFST, we use new threshold values on the Awerbuch's algorithm can be optimized to reduce to the communication complexity $O(n\sqrt{m})$ in sparse graphs whose graph density is less than 1.2. In the construction of DBFST using the new threshold values, the communication complexity becomes $O(\min(n\sqrt{m}, m + n^{1.6}))$.

Reference

1. B. Awerbuch and R. Gallager, A new distributed algorithm to find breadth first search tree, IEEE Trans. on Information Theory IT 33(3) (1987) 315–322.
2. B. Awerbach and R. Gallager, Distributed BFS algorithms IEEE Conf. Foundations of Computer Science, Portland, OR, (1985).
3. B. Awerbach, Complexity of network synchronization, JACM 32(4) (1985) 804–823.
4. T. Cheung, Graph traversal techniques and the maximum flow problem in distributed computation IEEE Trans. Software Engineering SE 9(4) (1983) 504–512.
5. G. Frederickson, A single source shortest path algorithm for a planar distributed network, Proc. 2nd Symp. Theoretical Aspects of Computer Science, Jan. 1985.
6. Y. Zhu, A new distributed breadth first-search algorithm, information Processing Letters, 25(10) (1987) 329–333.

✱ Appendix

Simulation results

A simulation is carried out to compare the communication complexities of the existing algorithms[1][3] and our proposed algorithm. The simulation is performed on SUN 3/50 system running under UNIX operating system. Test graphs with 50 nodes are used to measure the communication compleixty required to complete each algorithm and edges of the test graphs are connected randomly in each given graph density by using the system call, srand( ) (provided by UNIX system calls). In order to consider the non-detrministic property of communication in distributed environments, a random comunication speed is assiged to each link by using srand( ).

Three algorithms are excuted to obtain their communication complexitis on the test graph with a given graph density(between 1.0 and 2.0). Figure 1 shows the obtained results from the simulation. The x-axis shows the number of message to complete each algorithm and the y-axis represents the graph·density of a given test graph. The complexity of our proposed algoritrm approaches to the algorithm of[3] when graph density is less than 1.2 and comformed to the complexity of the algorithm of[1] when graph density is greater than 1.3. As a result, the proposed algorithm shows a better performance than the existing algorithms[1][3]. The algorithm using the graph density as threshold values, unifies the existing algorithm[1][3] and takes advantages of the existing algorithms selectively according to graph density.

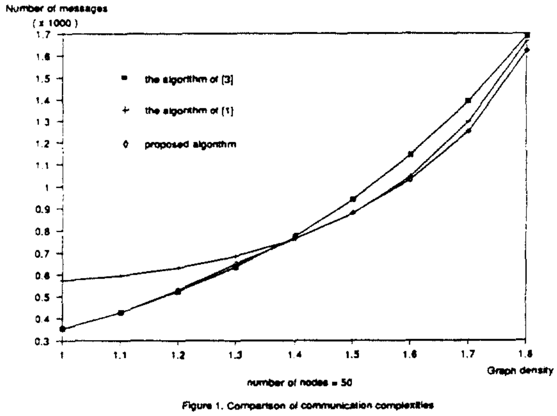

Figure 1. Comparison of communication complexities.

宋 仁 燮(In Sop SONG) 正會員
1952年 5月 15日生
1975年 2月：서울大學校 電子工學科 卒業 (工學士)
1977年～1979年：三星電子 綜合研究所 研究員
1980年～1989年：韓國機械研究所 研究員
1989年～現在：生産技術研究院 電氣電子 應用室 室長

申 宰 浩(Jae Ho SHIN) 正會員
1952年 8月29日生
1971年～1979年：서울大學校 工科大學 電子工學科卒業(工學士)
1980年～1982年：서울大學校大學院電子工學科 卒業(工學碩士)
1982年～1987年：서울大學校大學院電子工學科 卒業(工學博士)
1978年～1980年：大榮電子(株) 研究員
1983年～1988年：明知大學校 工科大學 電子工學科 助教授
1988年～現在：東國大學校 工科大學 電子工學科 副教授