Distributed synchronization algorithm for multi-agent system

著者	Ren Zhong, Yamamoto Shigeru
journal or publication title	Proceedings of the SICE Annual Conference
number	6060203
page range	1513-1516
year	2011-01-01
URL	http://hdl.handle.net/2297/29746

SICE Annual Conference 2011
September 13-18, 2011, Waseda University, Tokyo, Japan

Distributed Synchronization Algorithm for Multi-agent System

Zhong Ren[1] and Shigeru Yamamoto[2]

[1]Graduate School of Natural Science and Technology of Kanazawa University, Kanazawa, Japan
(Tel: +81-76-264-6332; E-mail: renzhong@moccos.ec.t.kanazawa-u.ac.jp)
[2]Faculty of Electrical and Computer Engineering of Kanazawa University, Kanazawa, Japan
(Tel: +81-76-234-4849; E-mail: shigeru@t.kanazawa-u.ac.jp)

Abstract: This paper provides a distributed algorithm to guarantee synchronization between agents for multi-agent systems. Motivated by vertex coloring from graph theory, we explore an approach based on tentative overlay as a condition mapping from interrelation and interaction between each agent, which equipped with local sensing and wireless communication capabilities. The objective of the proposed algorithms is to achieve synchronization, that is, making the most of cooperation of the agents in the multi-agent systems with network's connectivity, while other than nearest neighbor information, our approach assumes no knowledge of global network topology. We provide analysis and design results for multi-agent networks in arbitrary dimensions topology. The novel correctness poof relies on proximity graphs and their properties. In addition, simulations are provided that demonstrate the effectiveness of our theoretical results, for which we show a distributed dynamic programming of multi-agent system.

Keywords: Distributed control, Dynamic networks, Multi-agent systems.

1. INTRODUCTION

In recent years, decentralized coordination of multi-agent systems (MAS) has become an active area of research. A MAS is a system composed of many locally interacting and dynamically evolving agents, and the overall system will emerge some kinds of behaviors which will not be shown in a single agent, such as, phase transition, synchronization, clustering, pattern formation, swarm intelligence, etc. In the study of the collective synchronization, which is our target, is one of the behaviors. The tools used in the synchronization analysis now are matrix product and random geometric graphs.

In distributed computing by using wireless sensor networks, the smaller radius of communication makes the throughput of the distributed computing worse if the implemented distributed algorithm is itself slow. In general, it does not depend on the size of the network, but on the desired accuracy of the computation. On the other hand, one implication is that exchanging information via peer-to-peer network built on top of it will be extremely fast. Based on the above considerations, we consider a problem to create a multi-agent network where each agent can be touched by anyone else at the initial time. We present an algorithm to maximize the cooperation among agents which uses no knowledge of global network topology except for nearest neighbor information. The algorithm try to make nonadjacent agents be synchronized as possible as them can.

We derive an algorithm to obtain the number of colors by which synchronized agents are distinguished and the connectivity of the network is characterized in terms of relative positions and broadcast ranges R as communication capability. Moreover, nonadjacent agents are synchronized with the same color. Our derivation is motivated by the Welch-Powell algorithm [7] for vertex coloring in graph theory, which doesn't always yield a minimal coloring of G, and self-stabilizing algorithms in graphs as [8] in distributed computing. A distributed algorithm is said to be self-stabilizing if it can end up in a correct result no matter what is given as its initial state. We cast our problem into a vertex coloring problem which means color assignment to each vertex of a graph such that there is no edge connecting two identically colored vertices. Particularly, our algorithm only works for each agent with no knowledge of the whole system. We use the same color to display synchronized agents. The purpose is no two vertices sharing the same edge have the same color. This paper will emphasize analysis and correctness of the synchronization algorithm which given in our CCDC paper [1] in which the main part is the degree balance algorithm.

2. DEFINITION AND PROBLEM

2.1 The Topology of Multi-agent Systems

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Collective behavior of multi-agent systems is a significant point in the study of complex systems. Traditional topological theory is insufficient to describe a complex network such as a multi-agent system. This is because current complex networks emphasize the relationships between nodes and traditional topological theory is unable to define the relationships between nodes and describe an overall view of the network efficiently [2]. To our knowledge, this study is the first one to use geometric random graphs as in [3] to model the topology of multi-agent system. A example of multi-agent system is a wireless sensor network (WSN). WSN as in [4], [5] consists of spatially distributed autonomous sensors as three types of nodes (normal sensor nodes, aggregators, and a querier) to monitor physical or environmental conditions. The sensor nodes are usually scattered in a sensor field.

PR0001/11/0000-1513 ¥400 © 2011 SICE

2.2 Geometric Random Graphs

Definition 1 (Geometric Graphs [3]) A geometric graph $G(V, r) = (V, E, A)$ with radius r is a graph with node set V of points in a metric space, edge set E and adjacent matrix A.

Definition 2 (Geometric Random Graphs) Given a geometric graph $G = (V, E, A)$, a geometric random graph is a probability distribution over the set of all subgraphs of G.

According to Definition 1 and 2, we consider a dynamic multi-agent system with n agents in the two-dimensional space described by a weighted geometric random graph $G = (V, E, A)$, where a finite set of *vertices* $V = \{v_1, v_2, ..., v_n \mid v \in \mathbb{R}^2\}$, a set of ordered *edges* $E \subseteq V \times V$, and an adjacency matrix $A = [a_{ij}]$ whose entries $a_{ij} = 1$ if $(v_i, v_j) \in E$ and $a_{ij} = 0$ otherwise. Since we do not allow self-loops, for each i, $a_{ii} = 0$. The ith agent is assigned to node v_i. The edges $e_{ij} \in E$ represents the communication link. The communication capabilities give agents a potential communication bound, which is denoted by a circle centered on the agent i and given radius r_i. The radius represents communication capability of each agent. In this paper, we make an assumption that each agent has equal communication capability R, that is $r_i \equiv R$.

The multi-agents are equipped with sensors whose resolution is decaying exponentially with the distance to the object to observe. Hence, if the distance is less than or equal to the fixed connection radius R, then the agents are regarded as neighbors. When we define the pairwise distance between v_i and v_j as $d_{ij} = \|v_i - v_j\|$, the set of *neighbors* of

the agent i is denoted by

$$N_i = \{v_j \in V \mid 0 \le d_{ij} \le R\}. \tag{1}$$

We define a *subgraph* for agent i as a subgraph G_i of G whose vertices and edges from neighbors of agent i and the connected links. So, G_i can be seen in a circle with the radius R and with its center at the agent i. For example, in Fig. 1 is a graph G with 20 agents is depicted. The agent colored red is connected to neighbors that are covered by a circle with the radius R (which is depicted the red circle) in Fig. 1.

2.3 Connectivity

In our graph $G = (V, E, A)$, where A is the adjacent matrix, the Laplacian matrix of G, is denoted by L of which the rows and columns are indexed by V. The graph can be also represented using the Laplacian matrix:

$$L = D - A \tag{2}$$

where $D = \mathrm{diag}(d_1, d_2, ..., d_n)$ is a diagonal matrix with elements d_i. Obviously, the network is in spectral properties of complex networks with symmetric weights $A = A^T$. And then we know L is positive semi-definite and symmetric, its eigenvalues are all nonnegative. Let us denote the eigenvalues of A by

$$u_0 \ge u_1 \ge ... \ge u_{n-1} \tag{3}$$

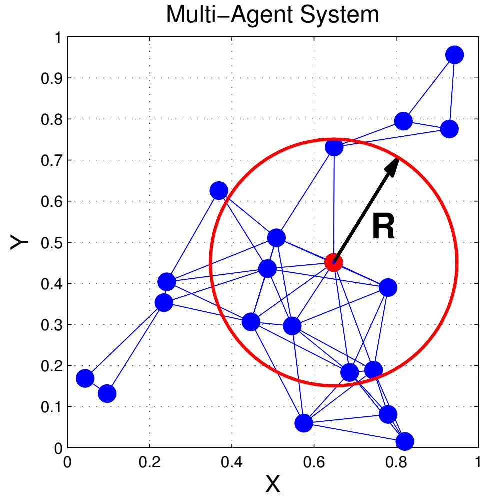


Fig. 1 Multi-Agent System with 20 agents

and by ordering the eigenvalues in a increasing way, we have the eigenvalues of L:

$$0 = \lambda_1 \le \lambda_2 \le ... \le \lambda_n \tag{4}$$

Lemma 1 (Algebraic Connectivity[6]) Let $\lambda_1(G) \le \lambda_2(G) \le ... \le \lambda_n(G)$ be the ordered eigenvalues of the Laplacian matrix $L(G)$. Then, $\lambda_1(G) = 0$, with corresponding eigenvector $\mathbf{1}$. Furthermore, $\lambda_2(G) > 0$ if and only if graph G is connected and hence $\lambda_2(G)$ is called the *algebraic connectivity* of G.

Letting $\mathfrak{C}_n$ be the set of all connected graphs include n agents, the graph which is connected is expressed as $G \in \mathfrak{C}_n$. Then, we make the assumption

Assumption 1 (G' Connectivity) Let $\lambda_2(G) > 0$ for all the time, so the $\forall G \in \mathfrak{C}_n$.
This assumption is presented for topology connectivity. $\lambda_2(G) > 0$ guarantee the given graph G connected forever.

2.4 System Design and Problem

We denote a matrix $C = (c_{ij}) \in \mathbb{R}^{n \times n}$, where $c_{ij} = 1$ if vertex i and j synchronized with the same color whereas $c_{ij} = 0$.

Proposition 1: Given graph $G = (V, E, A)$ with $|G| = n$, a vector $k(G)$-coloring of G is a row vector $c_j \in \mathbb{R}^n$ which is the j-th row of the matrix C_i corresponding to each vertex $i \in V$, such that for any two adjacent vertices i and v the inner product of their vectors satisfies

$$c_i^T c_v = \begin{cases} 0 \\ 1 \end{cases} \tag{5}$$

and for the maximum number of synchronized agents, it satisfies

$$\min k(G) \tag{6}$$

where $k(G)$ is all G for the same number of vertices n.

The definition of an orthonormal representation [10] [11] requires that the given dot products be equal to zero, a weaker requirement than the one above.

3. A SELF-ORGANIZING ALGORITHM FOR MULTI-AGENTS SYSTEMS

We describe Σ as a discrete-time system:

$$X(h+1) = X(h) + u(h) \quad (7)$$
$$C(h+1) = X(h) \wedge X(h)^T \quad (8)$$

where $h = 1, \dots, n$ denotes the discrete-time and X is a translation matrix of the output C which is a synchronization matrix as above mentioned, and $u(h)$ is a control matrix. Our objective is to find u to get final $C(n)$. The conjunction in Boolean algebra is denoted by $\wedge$.

Assumption 2 (Initial Assumption) We assume that all the agents are nonsynchronous, equally, $C(1) = I$ where I is the identity matrix. Moreover, $X(1) = I$.

3.1 Theoretical Algorithm Design

We consider an iterative algorithm which provides the final state $C(n)$ for a given initial $C(1)$. At first, we denote a reserve possible matrix $C' \in \mathbb{R}^{n\times n}$ as

$$C' = \overline{A+I} \quad (9)$$

where A is the adjacent matrix of the graph G and $\overline{A+I}$ means its elements are produced by the negation for $A+I$ as in Boolean algebra. Moreover, we wish to present a distributed algorithm, we denote a distributed matrix $B(h)$ and Bb as

$$B(h) = (C'P(h)) \wedge I = \mathrm{diag}(C'_{\cdot i}) \quad (10)$$

$$Bb = \overline{B(h)X(h)A_{\cdot h}}. \quad (11)$$

where $P(h) = e_h e_h^T \in \mathbb{R}^{n\times n}$ and e_h is the h-th column vector of the identity matrix with size n. For a distributed computing, we design the same rule for every agent i at the same step h

$$X_{\cdot i}(h+1) = X_{\cdot i}(h) + (C'_{\cdot i} \wedge Bb). \quad (12)$$

Then, the state $X(h)$ is updated by distributed decision of synchronization as

$$X(h+1) = X(h) + (C' \wedge Bb). \quad (13)$$

The above equations can be summarized as

$$X(h+1) = X(h) + (C' \wedge \overline{\mathrm{diag}(C'_{\cdot i})X(h)A_{\cdot h}}). \quad (14)$$

Hence, $u(h)$ in (7) is given as

$$u(h) = C' \wedge \overline{\mathrm{diag}(C'_{\cdot i})X(h)A_{\cdot h}}. \quad (15)$$

Note that this algorithm cannot get the smallest synchronization parameter $k(G)$ for the same scale n of the graph G. As we always know, vertex coloring problem is a nondeterministic polynomial problem and we are hard to figure out an excellent result. However, our notion and algorithm will improve operation rate cause of its distribution properties indubitably.

3.2 Vereinfacht Implementation Algorithm

Let different color be denoted by different RGB color number. We give a distributed algorithm which is Algorithm 1 involved in the initial color v_i and a set s_i for each agent i.

First, we define a vector V_i with n binary numbers as estimated queue using 0 or 1. We denote by 0 that the agent i is desynchronized by agent j, where $j \in \{ l \mid V_i(l) = 1\}$ and value 1 of the vector V_i means colored agents. On the other hand, we denote by 1 that the agent i is synchronized with the agent j. Let agent i's neighbor set N_i also translate into this 0,1 Boolean logical vector N_i^*, i.e., $n = 4, i = 1, N_i = \{2, 3\}$ so $N_i^* = [0, 1, 1, 0]$. Then, we present our algorithm following the function

$$V_i = \overline{N_i^*}. \quad (16)$$

Then, we put agent i itself into set s_i. When it contains 1 in V_i, we get $j := \arg\min\{l|V_i(l) = 1\}$ as a synchronous agent for agent i. The major view and methods of the algorithm design adopted in constructing synchronization sequence s_i through an alternative set V_i. Furthermore, i and j should be synchronized as:

$$V_i = V_i \wedge V_j, V_j = V_i. \quad (17)$$

In this moment, agent i and agent j have the same color $v_i = v_j$. And then, we put agent i, j into the set s_i, s_j separately for agent i and agent j.

Algorithm 1: A Distributed Algorithm for Agent i.

Input:
Subgraph $G_i = (V_i, E_i, A_i)$ getting from all $i \in \{1, 2, \dots, n\}$.
Output:
The color v_i and a synchronized set s_i of agent i.

First, we should give initial v_i with different color and $s_i = \phi$ for all agent i.
1. $s_i = \{i\}, V_i = \overline{N_i^*}$.
2. $V_i(s_i) = 0$.
3. if $\vee V_i(n) \neq 0$
    $j \in \{ l \mid V_i(l) = 1\}$.
   end if;
4. if $V_j(s_i) = 0$.
    $V_i = V_i \wedge V_j, V_j = V_i, v_i = v_j$.
   end if;
5. $s_i \leftarrow j, s_j \leftarrow i$

To analyze our algorithm, we give a theorem as follow:

Theorem 1 (Brooks' Theorem[9]) For any connected undirected graph G with maximum vertex degree $\Delta(G)$, the chromatic number of G which is the smallest number of colors $\mu(G)$ needed to color the vertices of G is at most Δ, that is, $\mu(G) \leq \Delta(G)$ unless G is a complete or an odd cycle (a cycle with an odd number of vertices).

For example, if a graph using 10 colors to synchronization finally, the chromatic number equal to 10. Bounds on the chromatic number is mentioned by Theorem 1 as above. It is proved for graph theory and tell us our algorithm is certainly convergent.

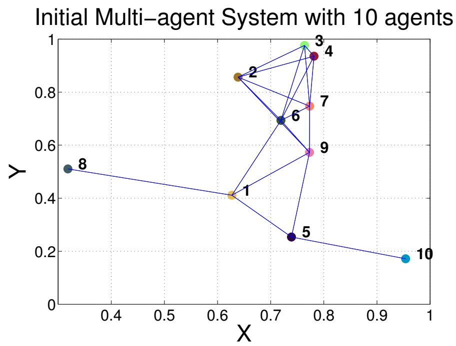
4. SIMULATION AND ANALYSIS

In this section, we provide some simulation results by our method. We applied Algorithm 1 to 10 agents in Fig. 2. The 10 agents with $R = 0.35$ are initially desynchronized and marked with the different 10 colors as shown in Fig. 2(a). By Algorithm 1, some of the 10 agents are synchronized which are illustrated with the same color as shown in Fig. 2(b) where the agents 1, 2, 10 are synchronized with color blue, the agents 3, 9 are pink, and the agents 5, 6, 8 are green. Meanwhile, Algorithm 1 guarantees neighbors asynchronism for all agents.

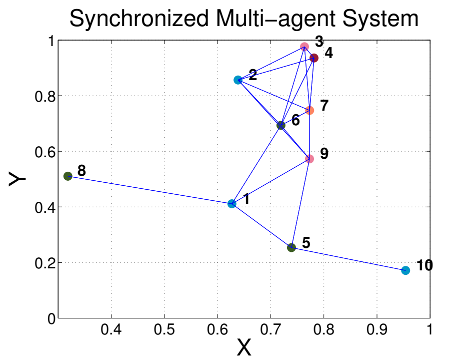
The algorithm has its superiority and inferiority. At first, there is always local results of algorithm, so it correspond with multi-agent system topology. The other is the algorithmic complexity is lower than global searching vertex coloring algorithm. And it can get local results quickly. However, the disadvantage is $k(G)$ is not the smallest one, even worse bigger.

5. CONCLUSION

We provides a distributed algorithm to guarantee synchronization between agents for multi-agent systems. Motivated by vertex coloring from graph theory, we explore an approach based on tentative overlay as a condition mapping from interrelation and interaction between each agent, which equipped with local sensing and wireless communication capabilities. While other than nearest neighbor information, our approach assumes no knowledge of global network topology. We provide analysis and design results for multi-agent networks in arbitrary dimensions topology. At last, simulations are provided that demonstrate the effectiveness of our theoretical results. For the future, we would like to combine our research into the dynamic system with connection changing or position moving.



(a)Initial Multi-agent System



(b)Synchronized System

Fig. 2 Distributed Synchronization Algorithm

REFERENCES

[1] Z. Ren and S. Yamamoto, "Distributed Connectivity Control in A Dynamic Network," Proc. of the 23rd Chinese Control and Decision Conference, pp. 2212–2217, 2011.

[2] H. L. Zhang, C. H. C. Leung and G. K. Raikundalia, "Classification of intelligent agent network topologies and a new topological description language for agent networks," Intelligent Information Processing III, IFIP International Federation for Information Processing (Shi, Shimohara, and Feng Eds.), Springer Boston, vol. 228, pp. 21–31, 2007.

[3] N. Przulj, D. G. Corneil and I. Jurisica, "Modeling interactome: scale-free or geometric?," Bioinformatics, Vol. 20, No. 18, pp. 3508–3515, 2004

[4] H. Alzaid, E. Foo, J. G. Nieto, "Secure data aggregation in wireless sensor network: a survey," Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference, pp. 315–320, 2006.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Communication Magazine, Vol. 40, No. 8, pp. 102–114, 2002.

[6] M. Fiedler, "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory," Czechoslovak Mathematical Journal, Vol. 25, No. 4, pp. 619–633, 1975.

[7] D. J. A. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," The Computer Journal, Vol. 10, No. 1, pp. 85–86, 1967.

[8] N. Guellati and H. Kheddouci, "A survey on self-stabilizing algorithms for independence, domination, coloring, and matching in graphs," Journal of Parallel and Distributed Computing, Vol. 70, No. 4, pp. 406–415, 2010.

[9] R. L. Brooks, "On coloring the nodes of a network," Mathematical Proceedings of the Cambridge Philosophical Society, Vol. 37, No. 2, pp. 194–197, 1941.

[10] L. Lovasz, "On the Shannon capacity of a graph," IEEE Transactions and Information Theory, Vol. 25, No. 1, pp. 1–7, 1979.

[11] M. Grotschel, L. Lovasz and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," Combinatorica, Vol. 1, No. 2, pp. 169–197, 1981.