数理解析研究所講究録
第 1941 巻 2015 年 41-44
2014 年度冬の LA シンポジウム [D-1]

Space Complexity of Self-Stabilizing Leader Election in Population Protocol on Hypernetworks

Xiaoguang Xu* Yukiko Yamauchi† Shuji Kijima‡ Masafumi Yamashita§

Abstract

Population protocol (PP) is a distributed computing model for passively mobile systems, in which a computation is executed by interactions between two agents. This paper investigates a generalized model, *population protocol based on interactions among at most k agents* (PP_k), where PP_2 is exactly the same as common PP model. Cai et al. (2012) showed that self-stabilizing leader election (SS-LE) in PP of n agents requires n agent states on *complete* communication network under the global fairness assumption, while Xu et al. (2013) gave a space complexity of SS-LE in PP_k, which is roughly $n/(k-1)$, as well. This paper shows that these space complexities are still sufficient for connected communication (hyper)graphs, in general. To be exact, our solution is not *static*, meaning that the leader agent is always unique but ever-changing after a convergence, for the purpose of self-stabilization avoiding deadlock. We also report an interesting fact that the space complexity of SS-LE in PP_k on some particular hypergraphs is strictly smaller than complete hypergraphs when $k \geq 3$.

1 Introduction

Population Protocol (PP), proposed by Angluin et al. [1], is a model of distributed systems consisting of mobile agents with limited computational resources, in which agent-to-agent communication (called interaction) is carried out only when two agents approach by accident where the model is motivated by networks such as networks of smart sensors attached to cars or animals, synthesis of chemical materials, complex biosystems, and so on (see also e.g., [3, 7]). Every agent is an identical finite state machine, and two interacting finite state machines (i.e., two agents under communication) can update their states by using a transition function. Once an initial configuration is given, an execution of the system is determined by the order of interactions among the agents, which however is unpredictable and is assumed to be given by an adversarial scheduler satisfying a *fairness condition.*

Population Protocol by interactions of at most k agents (PP_k) is a generalized model of PP where interactions among more than two agents are allowed. Such a generalization is already suggested by Angluin et al. [1], nevertheless very few facts, due to recent works [4, 11], are known on PP_k model.

The *Leader Election* (*LE*) is a problem to assign a special state of Q, representing the "leader", to exactly one agent. A configuration $C \in Q^n$ is *legitimate* if C contains exactly one agent in the leader state, and so does any configuration C' satisfying $C \xrightarrow{*} C'$. Let $\mathcal{L}(\subseteq Q^n)$ denote the set of all legitimate configurations. PP_k for LE is *self-stabilizing* (SS) (with respect to $\mathcal{L}$) if the following condition holds: For any configuration $C_0 \in Q^n$ and any execution $E = C_0 \xrightarrow{R_0} C_1 \xrightarrow{R_1} \dots$ starting from C_0, there is an $i \geq 0$ such that $C_i \in \mathcal{L}$. [1]

Angluin et al. [2] showed that there is no PP for SS-LE that works for any system of n agents, if n is not available to the agents. Fischer and Jiang [9] showed that there is a PP for SS-LE, if the scheduler is globally fair and the system can make use of the eventual leader detector $\Omega?$, that eventually detects the presence or absence of a leader. Canepa and Potop-Butucaru [6] proposed deterministic and probabilistic protocols when communication networks (i.e., interaction graphs) are rooted trees and arbitrary graphs, under the same assumption as [9]. Cai, Izumi, and Wada [5] returned to the original setting in [2] and asked a natural question:

*Graduate School of ISEE, Kyushu University
†Same as first author
‡Same as first author
§Same as first author

[1] We allow that the agent with leader state changes in successive configurations that appear after the system reaches a legitimate configuration.

how many agent states are necessary and sufficient in a PP for SS-LE? They then showed that n agent states are necessary and sufficient. Thus we cannot solve this typical and important problem in PP, unless we enhance the amount of agent memory until this large size, and this fact seems to contradict to the original design policy of PP. Xu et al. [11] extend the results to PP$_k$, and showed that the space complexity of SS-LE on PP$_k$ is exactly $\lceil (n-1)/(k-1) \rceil + 1$.

This paper shows that SS-LE in PP is solved with at most n agent states on any connected communication graph, which is exactly the same upper bound as the complete communication graph due to Cai et al. [5]. In fact, our solution is not *static*, meaning that the leader agent is always unique but ever-changing after a convergence. For general PP$_k$ ($3 \leq k \leq n$) of n agents, in a similar way, we show that the upper bound of the space complexity on an arbitrary appropriate communication hypergraph is $\lceil (n-1)/(k-1) \rceil + 1$, which is exactly the space complexity on "complete" communication hypergraph. What is interesting is that we show that when $k \geq 3$, the space complexity of SS-LE in PP$_k$ is strictly smaller on some communication hypergraphs than that of the complete communication hypergraphs.

2 Model Description

A population consists of n anonymous agents $u_1, u_2, ..., u_n$. Anonymous agents do not contain identifiers, and are treated in the same way during transition. An interaction graph $G = (V, E)$ is a simple undirected graph where each vertex representing an agent. The edge between u_i and u_j implies that agent u_i is able to interact with agent u_j. A population protocol based on k-interaction denoted with PP$_k$ ($k \geq 2$) is defined by (Q, δ), where $Q = \{q_0, q_1, \cdots, q_{m-1}\}$ denotes a finite set of states and an update function δ, defined by $Q^{k'} \rightarrow Q^{k'}$ maps each k'-tuple states to k'-tuple states ($2 \leq k' \leq k$). Notice that PP$_2$ is exactly the traditional PP. For PP$_k$ model, when the interaction graph is complete (denoted by complete PP$_k$), the scheduler is able to choose any k'-tuple of agents. For incomplete graph, the interaction graph specifies the possibilities of interactions among different agents. To allow full ability of interaction graph,we extend the common graph to hypergraph. And for a PP$_k$ model, there exists at least one hyperedge containing k agents.

A configuration is a n-tuple $(s_1, s_2, ..., s_n)$ of states with s_i corresponding to the state of agent u_i. In a configuration, it is able to have more than one agent with the same state q. A transition from a configuration C to the next configuration C' in an PP$_k$ is defined as follows. At the beginning, the scheduler chooses a k'-tuple ($2 \leq k' \leq k$) of agents $(u_1, u_2, ..., u_{k'})$. Suppose the states of the k'-tuple agents are $(s_1, s_2, ..., s_{k'})$ respectively, and let R: $(s_1, s_2, ..., s_{k'}) \rightarrow (s'_1, s'_2, ..., s'_{k'})$ be a transition rule of δ. Then, the k'-tuple agents $(u_1, u_2, ..., u_{k'})$ interact, denoted by $C \overset{R}{\rightarrow} C'$, and the states of agents $(u_1, u_2, ..., u_{k'})$ in C' are $(s'_1, s'_2, ..., s'_{k'})$ respectively, while all other agents keep their states in the transition. We say that a transition $C \overset{R}{\rightarrow} C'$ is *active* if at least one agent changes its state or *silent* otherwise.

An execution E is defined as an infinite sequence of configurations and transitions in alternation $C_0, R_0, C_1, R_1, \ldots$ such that $C_i \overset{R_i}{\rightarrow} C_{i+1}$ for each i. Like most of the literature on PP, we assume that the scheduler in each of models in this paper is adversarial, but satisfying some *fairness* conditions.

A scheduler is said to be *strongly fair* while following the condition that if C is a configuration that appears infinitely often in an execution, and there exists Rsuch that $C \overset{R}{\rightarrow} C'$, then C' must also appear infinitely often in the execution. All protocols given in this paper are assumed to run under global fairness.

We sometimes abbreviate $C \overset{R}{\rightarrow} C'$ to $C \rightarrow C'$, unless it is confusing. The reflexive and transitive closure of $\rightarrow$ is denoted by $\overset{*}{\rightarrow}$. That is, $C \overset{*}{\rightarrow} C'$ means that a configuration C' is reachable from a configuration C by a sequence of transitions of length greater than or equal to 0.

We show that SS-LE on any general graph has the same upper bound of space complexity as space complexity on complete hypergraph. Considering the limited resources on mobile agents, we are concerned with finding special hypergraphs which can reduce the lower bound of space complexity. And we introduce the following two cases:

PP$_{k_l}$: n agents in total are separated into two groups S and $\overline{S}$. With the given number k and l ($2 \leq l \leq k-1$), the scheduler chooses k' ($2 \leq k' \leq k$) agents in total from the groups S and $\overline{S}$ in each interaction with one condition that at most l agents are chosen from $\overline{S}$ in any interaction. Group $\overline{S}$ is guaranteed to have more than l agents, otherwise the condition makes no sense.

PP$_{k*c}$: n agents are separated into c groups. Each group is guaranteed to have at least k agents. The scheduler chooses k' ($2 \leq k' \leq k$) agents in total from all the groups in each interaction with one condition that at most $k-1$ agents are chosen from each group.

3 General Graphs

This section considers the upper bound of space complexity of PP_k for SS-LE on general hypergraphs.

3.1 Upper Bound of the Space Complexity of General PP

General PP is PP on a general graph. The following theorem presents an upper bound of the space complexity of general PP for SS-LE.

Theorem 1. *There exists a PP using n agent-states which solves the SS-LE for n agents in any general network.*

Proof. We present the following protocol 1.

Protocol 1. $Q = \{q_0, q_1, \ldots, q_{n-1}\}$, *where q_0 denotes the leader state.*
$\delta = \{$
$R_1: (q_i, q_i) \to (q_{i-1 \pmod n}, q_i)$, *for any* $i \in 0, \ldots, n-1$,
$R_2: (q_i, q_j) \to (q_j, q_i)$, *for any* $i \neq j$.
$\}$

In protocol 1, if rule R_1 is applied on a complete network, agents in the same state are not able to coexist. And finally each agent is in a distinct state as proved by Cai, Izumi, Wada [5]. On a general network, we add an additional rule R_2 to make sure agents in the same state be able to interact without affecting the size of agents in any state. Finally each agent will be in a distinct state. Then the configuration contains exactly one leader agent and the size of agents in each state is not able to change any more according to all the rules. The effect of R_2 is that while agents u_i and u_j in the same state q are not adjoined to each other, there must exist a path connecting agents u_i and u_j. Agent u_j is able to exchange the states with the neighbor on the path that is closer to u_i. By exchanging the states one by one towards u_i, we are able to apply R_1 on state q. If there exists one agent in state q between u_i and u_j, they can interact as well because the goal is to interact with an agent with the same state. In this way, we are able to get an execution $C \overset{*}{\to} C'$ where C is an arbitrary configuration and $C' \in \mathcal{L}$ is a legitimate configuration that each agent stays in a distinct state. Since then the number of agents in each state is not able to change, we obtain that for each $C' \to C''$, $C'' \in \mathcal{L}$. □

3.2 Upper Bound of the Space Complexity of General PP_k

The following theorem presents an upper bound of the space complexity of general PP_k for SS-LE.

Theorem 2. *There exists a PP_k using $\lceil (n-1)/(k-1) \rceil + 1$ agent-states which solves the SS-LE for n agents on any general network.*

Proof. We present the following protocol 2 in case $n \equiv 1 \pmod{k-1}$.

Protocol 2. $Q = \{q_0, q_1, \ldots, q_{m-1}\}$, *where q_0 denotes the leader state,* $m \overset{\text{def}}{=} (n-1)/(k-1) + 1$ *denotes the size of agent-states.*
$\delta = \{$
$R_1: (q_0, \ldots, q_0, q, \ldots, q') \to (q_0, q_{m-1}, \ldots, q_{m-1}, q, \ldots, q')$, *for any* $q, \ldots, q' \in Q \setminus \{q_0\}$,
$R_2: (q_i, q_i, \ldots, q_i) \to (q_{i-1}, q_i, \ldots, q_i)$, *for any* $i \in \{1, 2, \ldots, m-1\}$,
$R_3: (q, q', \ldots, q'') \to (q', \ldots, q'', q)$, *in cases other than R_1 and R_2.*
$\}$

In protocol 2, if rule R_1 and R_2 are applied on a complete network, under global fairness, we are able to reach a configuration where there exists exactly one agent in state q_0 and $k-1$ agents in any other state, as showed by Xu et al. [11]. The idea is similar to general PP. Since there exists at least one hyperedge adjoining k agents, we can transfer any state onto this hyperedge and apply R_1 or R_2 to change the states. In case there exists more than one agent in leader state or more than $k-1$ agents in any other state, we are able to eliminate one of them.
Proof in case other than $n \equiv 1 \pmod{k-1}$ is similar that we omit here. □

4 Reducing Space Complexity on Special Graphs

This section introduces two special graphs where the space complexity can be reduced comparing to complete graph. The following theorems present upper bounds of the space complexity of $PP_{k\perp}$ and PP_{k*c} ($k \geq 3$) for SS-LE. We omit the proofs due to page limitation.

Theorem 3. *For any integer k ($k \geq 2$), and for any integer n ($n = |S| + |\overline{S}| \geq k$), there exists a $PP_{k\perp}$ using $\lceil (n - |\overline{S}| - 1)/(k-1) \rceil + 2$ agent states ($|\overline{S}| > l$) which solves the SS-LE for n agents.*

Theorem 4. *For any integer k ($k \geq 3$), and for any integer c, there exists a PP_{k*c} using $\lceil (min(|U_1|, ..., |U_c|) - 1)/(k-2) \rceil + c$ agent states which solves the SS-LE for n agents.*

5 Emulation of SS Algorithms on PP_k

This section emulates SS algorithms on a complete graph to any general graph. Emulations in this section have a restriction, the legitimate configuration does not have requirement on positions of agents in any state. The following theorems present an emulating PP and PP_k of SS algorithm solving problems under a complete graph. We omit the proofs due to page limitation.

Theorem 5. *Under PP model, if on a complete graph, there exists an SS algorithm A solving a problem with m states, then on any general graph, there exists an algorithm A' solving the problem with $2m$ states under global fairness.*

Theorem 6. *Under PP_k model, if on a complete graph, there exists an SS algorithm A solving a problem with m states, then on any general graph, there exists an SS algorithm A' solving the problem with $2m$ states under global fairness.*

6 Conclusion

This paper first gives a PP and PP_k solving SS-LE on any general graph with the upper bound of space complexity exactly the same as a complete graph. Then investigates two new models. One is called $PP_{k\perp}$ and we give an upper bound of space complexity solving SS-LE. Another is called PP_{k*c} and we give two upper bounds of space complexity solving SS-LE in case $k \geq 3$. Still we are not familiar with the lower bounds of these problems currently, and would like to consider in the future. At last we provide emulating PP and PP_k of SS algorithms solving problems on a complete graph.

References

[1] Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M.J., Peralta, R.: Computation in networks of passively mobile finite-state sensors. Distributed Computing, **18** (2006), 235–253.

[2] Angluin, D., Aspnes, J., Fischer, M.J., Jiang, H.: Self-stabilizing population protocols. ACM Transactions on Autonomous and Adaptive Systems, **3** (2008), Article 13.

[3] Aspnes, J., Ruppert, E. An introduction to population protocols, Bulletin of the EATCS, **93** (2007), 98–117.

[4] Beauquier, J., Burman, J., Rosaz, L., Rozoy, B.: Non-deterministic population protocols, LNCS, **7702** (2012), 61–75.

[5] Cai, S., Izumi, T., Wada, K.: How to prove impossibility under global fairness: on space complexity of self-stabilizing leader election on a population protocol model, Theory of Computing Systems, **50** (2012), 433–445.

[6] Canepa, D., Potop-Butucaru, M.G.: Stabilizing leader election in population protocols. INRIA Rocquencourt, RR-6269, 2007. available from `http://hal.inria.fr/inria-00166632/en/`

[7] Chatzigiannakis, I., Michail, O., Spirakis, P.G.: Recent advances in population protocols. Lecture Notes in Computer Science, **5734** (2009), 56–76.

[8] Chatzigiannakis, I., Michail, O., Spirakis, P.G.: Mediated population protocols. Theoretical Computer Science, **412** (2011), 2434–2450.

[9] Fischer, M.J., Jiang, H.: Self-stabilizing leader election in networks of finite-state anonymous agent. Lecture Notes in Computer Science, **4305** (2006), 395–409.

[10] Mizoguchi, R., Ono, H., Kijima, S., Yamashita, M.: On space complexity of self-stabilizing leader election in mediated population protocol, Distributed Computing, **25** (2012), 451–460.

[11] Xu, X., Yamauchi. Y, Kijima, S., Yamashita, M.: Space Complexity of self-stabilizing leader election in population protocol based on k-interaction, LNCS **8255**, (SSS 2013), 86–97.