International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 9, Number 2, pages 239–256, July 2019

Energy Balancing by Wireless Energy Transfer in Sensor Networks

Hiroshi Michizu
Graduate School of Information Science and Technology, Osaka University,
h-michizu@ist.osaka-u.ac.jp
Yamadaoka 1, Suita, 565-0871, JAPAN

Yuichi Sudo
Graduate School of Information Science and Technology, Osaka University,
y-sudou@ist.osaka-u.ac.jp
Yamadaoka 1, Suita, 565-0871, JAPAN

Hirotsugu Kakugawa
Faculty of Science and Technology, Ryukoku University,
kakugawa@rins.ryukoku.ac.jp
Yokotani, Seta Oe-cho 1-5, Otsu, 520-2194, JAPAN

Toshimitsu Masuzawa
Graduate School of Information Science and Technology, Osaka University,
masuzawa@ist.osaka-u.ac.jp
Yamadaoka 1, Suita, 565-0871, JAPAN

Received: February 15, 2019
Revised: May 6, 2019
Accepted: June 5, 2019
Communicated by Akihiro Fujiwara

Abstract

Wireless energy transfer is a technology to transmit electricity without wire, and it is a promising technology for charging battery of mobile devices. In battery powered sensor networks, it is important to balance electric energy of batteries of nodes in order to maximize the life time of networks. In this paper, we propose three distributed protocols to balance electric energy of batteries of nodes. The proposed algorithms are based on the population protocol model which is a computational model for networked nodes with very limited resources. The goals of the algorithms are twofold: minimizing the loss of electric energy caused by wireless transmission, and minimizing the time to balance. The proposed algorithms are evaluated by computer simulation.

Keywords: distributed system, sensor networks, wireless energy transfer, population protocol

1 Introduction

In recent years, sensor networks that consist of a large number of tiny sensor nodes equipped with wireless communication devices are widespread [2]. Sensor nodes are often powered by small batteries, and they will stop operation when battery power is exhausted. Therefore, in order to keep the network in operation for a long time, it is important to save the power consumption of each node and balance the remaining power of the nodes.

There are two directions for balancing the battery power in literature. One direction is adjusting loads of communication and computation such that nodes with more battery power have higher loads. A lot of research are devoted in this direction. Another direction is to balance battery power by wireless energy transfer such that nodes with more power transfers energy to nodes with less power by transmitting and receiving electro-magnetic waves. This direction is an emerging research area in recent years.

In this paper, we aim at balancing the remaining battery power of nodes by wireless energy transfer. To this end, we propose distributed protocols based on the population protocol model [1] which is one of computational models for wireless mobile sensor networks with limited computational resources. The nodes in the population protocol model are called agents, and the network is called a population. When two agents are sufficiently close to each other, an interaction occurs between the agents, and the states of the two agents change according to the protocol. We assume that each nodes is equipped with wireless energy transmitter and receiver, and electric power can be transmitted and received between two agents in the interaction. However, when transmitting and receiving power, some fraction of power is lost. Whenever agent u transfers energy x to agent v, agent v receives energy $(1 - \beta)x$, where β is the loss factor which depends on the environment and the energy transmitter and receiver of agents. Therefore, reducing the loss in energy transmission is an important issue to attain long life of sensor networks in addition to balancing the power of all the agents. The aim of this paper is to design protocols that achieve the following two goals. (1) Small loss of energy transmission. (2) Fast convergence time to balance the battery power of nodes.

As a related work [6], the authors propose several balancing protocols, and evaluate the protocols by computer simulation. One of their protocols is based on the idea that the amount of energy to be transmitted is decided in such a way that the amount of remaining batteries of sender and receiver agents are equalized. A generalization of this protocol is proposed in this paper. In [4], protocols are proposed to balance battery power of sensor networks of star topology such that energy of the center node and the sum of the remaining nodes are equal, and the protocols are compared each other.

In this paper, we propose three protocols in the population model to balance the remaining power of agents. One is a local protocol whose transmission power is determined only by the remaining power of two interacting agents at the time of interaction. Another is a global protocol that estimates the total number of agents and the total remaining power at the initial state in the population, and based on this estimation, interacting agents determine the amount of transmission power. The other is also a global protocol that estimates two agents that have maximum or minimum remaining power in the population, and based on this estimation, only these two agents exchange power. These three protocols are evaluated by computer simulation experiments with two metrics of execution time and power loss rate.

The organization of this paper is as follows. First, in Section 2, we present the population protocol model and introduce the model of wireless power transmission. Next, in Section 3, we propose three power balancing protocols. In Section 4, we present evaluation results of the proposed protocols by computer simulation. Finally, in Section 5, we present a summary and future work.

2 Definitions

In this section, we present the definition of the population protocol model with the transmission and reception ability of electric power, and formulate the energy balancing problem. The population protocol model is often used to abstract sensor networks. Several definitions follow [4].

2.1 Population Protocol

We consider a population of n agents represented by a graph $G = (V, E)$, where $V = \{u_1, u_2, \ldots, u_n\}$ is the set of agents, and $E \subseteq V \times V \setminus \{(v, v) \mid v \in V\}$ is the set of edges in the population. Each edge $(u, v) \in E$ indicates the possibility that the agents u and agent v interact with each other as an initiator and a responder respectively.

Each agent is equipped with a battery cell, a wireless energy transmitter, a wireless energy receiver, and a local memory consisting of a small number of registers. We assume that agents are identical, that is they do not have unique IDs, they have the same hardware and run the same protocol.

For any time $t \geq 0$, the state of the agent u at time t is defined as follows.

$$C_u(t) = (W_u(t), R_u(t))$$

Here, $W_u(t)$ (resp., $R_u(t)$) is the remaining energy (resp., the local memory) of agent u at time t. Whenever agents u and v interact, they update local memory and may transmit energy according to protocol P. If agents u and v interact with each other as an initiator and a responder respectively at time t, they modify their states as follows:

$$(C_u(t+1), C_v(t+1)) = P(C_u(t), C_v(t))$$

Notice that, as we will explain in Section 2.2, there is a restriction in the relationship between the remaining energy (i.e., $W_u(t)$ and $W_v(t)$) at time t before interaction and those (i.e., $W_u(t+1)$, $W_v(t+1)$) at time $t+1$ after interaction. It is assumed that the number of interactions that take place is exactly one at each time t, and the states of other agents that do not join the interaction are unchanged. The interaction performed by two agents is asymmetric because one of the two agents plays the role of the initiator and the other plays the role of the responder. In the following, the function $C_u(t)$ that gives the state of each agent at time t is called the state of the population at time t. In particular, $C_u(0)$ is called the initial state.

2.2 Transmit and Receive Energy

Whenever a pair of agents interacts, they are able to exchange energy by using their wireless energy equipment. However, energy transmission incurs energy loss [4]. Whenever an agent u transmits energy of amount x to agent v, the amount of energy that agent v actually receives is $(1-\beta)x$, where $\beta \in [0, 1)$ is a loss ratio depending on the environment and the equipment for energy transfer available to the agents. Therefore, when agents u and v interact at time t and u transmits energy x to v, the new energy levels of u and v at time $t+1$ are represented as follows:

$$W_u(t+1) = W_u(t) - x \tag{1}$$

$$W_v(t+1) = W_v(t) + (1-\beta)x \tag{2}$$

We assume that each agent knows the value of β.

2.3 Balancing Energy

In this paper, we propose three protocols for balancing the energy of agents. In the initial state, each agent has an arbitrary amount of energy. For each time t, let $W_{max}(t)$ and $W_{min}(t)$ be the maximum and minimum energy of agents at t respectively. We say that the energy distribution at time t is $(1+\epsilon)-$ balancing if $\frac{W_{max}(t)}{W_{min}(t)} \leq 1+\epsilon$ is satisfied. The execution time t_{fin} of the protocol is defined to be the minimum time t such that the energy distribution becomes $(1+\epsilon)-$balancing for the first time from the initial state. Recall that exactly one interaction takes place at each time t, and the execution time is equal to the number of interactions until the energy distribution becomes $(1+\epsilon)-$balancing. To the best of our knowledge $W_{max}(t)$ and $W_{min}(t)$ is monotonically non-increasing and non-decreasing respectively in all the protocols. This implies that the property of $(1+\epsilon)-$balancing is kept once it is attained. The energy loss rate is defined as follows:

$$\frac{\sum_{i=1}^{n} W_{u_i}(0) - \sum_{i=1}^{n} W_{u_i}(t_{fin})}{\sum_{i=1}^{n} W_{u_i}(0)}$$

2.4 Scheduler

The scheduler is a virtual entity that determines two agents that interact at each time. We assume the uniformly random scheduler, that is, every (ordered) pair of agents in E has the same probability to be selected at each time. This scheduler is represented by a sequence of interactions $\Gamma = \gamma(0), \gamma(1), \ldots$, where each interaction $\gamma(t)$ is random variable such that $\Pr(\gamma(t) = (u, v)) = \frac{1}{|E|}$ for each $t \geq 0$ and $(u, v) \in E$.

3 The Proposed Protocols

In this section, we propose three protocols P_{BLA}, P_{BGA}, and P_{BMM} for achieving $(1+\epsilon)-$balancing of the population for any given ϵ. We present the details how to compute the amount of energy transmission.

3.1 Protocol Based-On-Local-Average P_{BLA}

When agents u and v interact, the protocol P_{BLA} determines the amount x of energy transmission based on the amount that should be transfered to make u and v have the same amount of energy. The agent with larger energy transmits the amount x of energy to the other agent. In this protocol, the amount of energy transmission is determined only from the remaining energy of the interacting agents. In case two agents have the same amount of energy when they interact, energy transmission is not performed.

First, let us explain how to compute the amount x of energy transmission from agent u to v. Without loss of generality, we assume that $W_u(t) \geq W_v(t)$. Since the energy of agent u and v after the interaction change according to the equations (1) and (2), the transmission amount x which satisfies $W_u(t+1) = W_v(t+1)$ is as follows.

$$x = \frac{W_u(t) - W_v(t)}{2 - \beta}$$

In this protocol P_{BLA}, the actual amount x' of energy transmission from agent u to v is determined from x and a parameter of a positive real number $\alpha(\geq 1)$:

$$\begin{aligned} x' &= \frac{x}{\alpha} \\ &= \frac{W_u(t) - W_v(t)}{\alpha(2 - \beta)} \end{aligned}$$

The reason agent u adopts x' instead of x as the amount of energy transmitted to agent v comes from the following observation. If α is small, the transmission amount per one intersection is large, so it can be expected that $(1+\epsilon)-$balancing will be achieved in a short time. However, when energy is transmitted and received between agents with large (resp., small) energy, agent that have received (resp., have transmitted) energy need to transmit (resp., receive) energy later. So, the amount of energy of agents is likely to oscillate, and the energy transfer is considered to be wasteful due to the loss of wireless energy transfer. If α is small, the total amount of energy loss due to this wasteful energy transfer is expected to increase. For example, suppose that agent u with energy of 100 and agent v with 20 are interacting in a situation where the average value of energy of agents is 20. In the case of $\alpha = 1$ and $\beta = 0.4, 50$ energy are transmitted from agent u to v, v receives 30 energy and the amount of v's energy are increased to 50. However, since the energy of every agents must converge to 20 or less, agent v must transmit 20 or more energy to other agents. Thus, energy transmission of 50 from agent u to v worsens the situation not only losing 20 energy, but also away from $(1+\epsilon)-$balancing convergence value. If α is increased, the execution time is expected to linearly increase, but on the other hand, it is expected that the amount of energy loss due to wasteful energy transfer as described above decreases. This implies that if α is increased to the limit, the energy of an agent having larger (resp., smaller) energy than the average energy of all agents almost monotonically decreases (resp., increases) to the final value.

The pseudocode of the protocol P_{BLA} is shown in Protocol 1.

Protocol 1 P_{BLA}

```
$x \leftarrow |\frac{W_u(t)-W_v(t)}{\alpha(2-\beta)}|$
**if** $W_u(t) \geq W_v(t)$ **then**
    $W_u(t+1) \leftarrow W_u(t) - x$
    $W_v(t+1) \leftarrow W_v(t) + (1-\beta)x$
**else**
    $W_u(t+1) \leftarrow W_u(t) + (1-\beta)x$
    $W_v(t+1) \leftarrow W_v(t) - x$
**end if**
```

3.2 Protocol Based-On-Global-Average P_{BGA}

This protocol P_{BGA} first elects a leader, and the leader estimates the total number of agents and the total amount of energy in the initial situation. The leader uses these values to estimate the average amount of energy in the initial situation. Based on the estimated average, we also estimate the upper and lower limits of the finally balanced energy of agents. The idea of this protocol is to avoid wasteful energy transmission and reception between two agents when they interact if both of them have energy larger (resp., smaller) than the upper (resp., lower) limit. This idea is based on the observation that energy transmission and reception between two agents both having exceeding (resp., insufficient) energy never contribute to balancing of energy of all agents.

Each agent has the following variables.
· *leader*: Indicates whether it is a leader, 1 for a leader, 0 for a non-leader (Initial value is 1)
· *num*: Number of agents estimated by the leader (Initial value is 1)
· W_{total}: Total energy estimated by the leader (Initial value is its own initial energy)
· *count*: The number of times the leader has continuously interacted with non-leaders (Initial value is 0)
· num_{mid}: The maximum value of num (Initial value is 1)
· num_{final}: Estimated number of agents in population (Initial value is 1)
· W_{ave}: Estimated average amount of energy in the initial situation (Initial value is 0)
· *border*: Threshold for judging that there is only one leader in population (The value is determined by preliminary experiments)

The pseudocode of P_{BGA} is shown in Protocol 2.

Protocol 2 P_{BGA}

```
$ElectLeader()$
$Broadcast()$
$TransmitEnergy()$
```

This protocol consists of three functions $ElectLeader()$, $Broadcast()$, and $TransmitEnergy()$ as shown in Protocol 2, and every agent executes them in this order that is, the protocol consists of the following three phases:
- Leader Election Phase
- Broadcasting Phase
- Transfer Phase

At each interaction, all phases are performed in this order. However, in Broadcasting Phase, the estimated values are not broadcasted if it is not judged that only one leader exists and in Transfer Phase, energy transfer is not performed if the average amount of energy in the initial situation is not estimaed.

In Leader Election Phase, which is described in function *ElectLeader*(), a unique agent is elected as the leader of all agents. The pseudocode of this phase is shown below: agents u and v interact with u as an initiator and v as a responder. The symbol "↔" in the pseudocode means swapping the values of two variables.

```
ElectLeader()

if ($leader_u = leader_v = 1$) then
    //processing of an initiator
    $num_u \leftarrow num_u + num_v$
    $W_{total_u} \leftarrow W_{total_u} + W_{total_v}$
    $count_u \leftarrow 0$
    //processing of a responder
    $leader_v \leftarrow 0$
    $num_v \leftarrow 0$
    $W_{total_v} \leftarrow 0$
    $count_v \leftarrow 0$
else if ($leader_u = 1 \wedge leader_v = 0$) $\vee$ ($leader_u = 0 \wedge leader_v = 1$) then
    $leader_u \leftrightarrow leader_v$
    $num_u \leftrightarrow num_v$
    $W_{total_u} \leftrightarrow W_{total_v}$
    $count_u \leftrightarrow count_v$
    let $l$ be the agent ($u$ or $v$) with $leader_l = 1$, and $f$ be the other
    $count_l \leftarrow count_l + 1$
    if $num_l < max(num_{mid_f}, num_{mid_l})$ then
        $count_l \leftarrow 0$
    end if
end if
$num_{mid_u} \leftarrow max(num_{mid_u}, num_{mid_v}, num_u, num_v)$
$num_{mid_v} \leftarrow max(num_{mid_u}, num_{mid_v}, num_u, num_v)$
```

Initially, each agent is a leader candidate, and when two leader candidates interact, only one of them continues to be a leader candidate. Eventually, exactly one leader is elected. Below, we call an agent that is a leader candidate simply as a leader. In addition to electing a unique leader in this phase, each leader counts the number of agents in variable of *num*, and sums up the amount of initial energy of them in variable W_{total}. When a unique leader is elected, it holds the total number of agents in variable *num* and the total initial energy of all agents in variable W_{total}. Of course, the value of W_{total} is just an upper bound but it is important information to achieve balancing state. Variable *count* is used to determine whether a unique leader has been elected or not. If a leader interacts with a non-leader, the value of *count* is incremented by 1, and if it interacts with a leader, the value of *count* is reset to 0. When the value of *count* exceeds a threshold value, a leader decides that there is no other leader and it is unique. The value of num_{mid} indicates the maximum value of *num* observed so far. The value of num_{mid} is shared by agents: when an interaction takes place, the larger value spreads. That is, at each interaction, each agent sets the value of its num_{mid} to the maximum of values of *num* and num_{mid} of the two agents. When a leader observes that its value of *num* is less than the value of num_{mid} of the interacting agent, its value of *count* is reset to 0 because this situation means that there exists another leader having a larger value of *num* in the population.

The pseudocode describes actions of two agents at an interaction. Let u be the initiator and v be the responder at the interaction.
1. An interaction by two leaders (Lines 1 to 10 and 22 to 23):
Agent u receives the values of num and W_{total} from agent v and adds the values to its values of num and W_{total}. On the other hand, v becomes a non-leader, and the values of num, W_{total} and leader are set to 0. Also the value of count of u is reset to 0. The value of num_{mid} each agent has is set to the maximum value of num_{mid} of the two agents and value of num each agent has.
2. An interaction by two non-leaders (Lines 22 to 23):
The values of num, W_{total}, and leader of the two non-leaders are not changed. The value of num_{mid} of each agent is updated.
3. An interaction by a leader and a non-leader (Lines 11 to 23):
The values of num, W_{total}, and *leader* of the two agents is swapped respectively. This operation is needed to make sure that the leader interacts with every other agent even if the interacting network is not a complete graph. The value of *count* of a leader is incremented by 1. The value of num_{mid} of each agent is updated. If one of the values of num_{mid} of two agents is larger than the value of num of a leader, the value of *count* of a leader is reset to 0.

In Broadcasting Phase, which is described in function *Broadcast*(), the unique leader estimates the average amount of energy in the initial state, and broadcasts it to all agents. Note that energy transfer is not actually done in this phase, it is executed in the next phase. The pseudocode of this phase is shown below.

Broadcast()

```
if $(count_u > (num_u + 1) * border) \vee (count_v > (num_v + 1) * border)$ then
    if $count_u > (num_u + 1) * border$ then
        $num_{final_u} \leftarrow num_u$
        $num_{final_v} \leftarrow num_u$
        $W_{ave_u} \leftarrow W_{total_u}/num_u$
        $W_{ave_v} \leftarrow W_{total_u}/num_u$
    else
        $num_{final_u} \leftarrow num_v$
        $num_{final_v} \leftarrow num_v$
        $W_{ave_u} \leftarrow W_{total_v}/num_v$
        $W_{ave_v} \leftarrow W_{total_v}/num_v$
    end if
end if
if $(num_{final_u} > 1) \vee (num_{final_v} > 1)$ then
    if $num_{final_u} > num_{final_v}$ then
        $num_{final_v} \leftarrow num_{final_u}$
        $W_{ave_v} \leftarrow W_{ave_u}$
    end if
    if $num_{final_u} < num_{final_v}$ then
        $num_{final_u} \leftarrow num_{final_v}$
        $W_{ave_u} \leftarrow W_{ave_v}$
    end if
end if
```

The timing to start broadcasting here is determined using variable of *border*. In the simulation experiments of Section 4, the value of *border* is determined from 100 preliminary experiments so that only one leader exists in most cases (more than 99 percent of cases). So, by using this value, correct estimated values are eventually broadcasted in most cases. Agents repeat interactions, and when a leader finds that the value of *count* exceeds the threshold, then it decides that there is only one leader in the population (Line 1). Note that this decision may not be correct. When a leader decides that it is unique (and if it is correct), the value of num of the leader is the total number of agents in the population and the value of W_{total} is the total amount of the initial energy of agents. So the average amount of the energy in the initial situation is estimated by an expression W_{total}/num. The number of agents in the population is stored in num_{final} and the average amount of the energy in the initial situation is stored in W_{ave} (Lines 2 to 12). When agent u and agent v interact, agent u (resp., agent v) passes its own estimated value to agent v (resp., agent u) if the estimated value agent u has is larger (resp., smaller) than the one agent v has (Lines 14 to 23).

In Transfer Phase, which is described in function $SendEnergy()$, agents actually transmit and receive energy, and the amount of energy transmission is decided based on the estimated value of the average amount of energy in the initial situation. The pseudocode of this phase is shown below.

SendEnergy()

```
 1: if $(num_{final_u} > 1) \land (num_{final_v} > 1)$ then
 2:     $upper \leftarrow (1-\beta)W_{ave_u} * U$
 3:     $lower \leftarrow (1-\beta)W_{ave_u} * L$
 4:     if $W_u(t) \geq W_v(t)$ then
 5:         $x \leftarrow max\left(0, min(\frac{W_u(t)-W_v(t)}{2-\beta}, \frac{upper-W_v(t)}{1-\beta}, W_u(t)-lower)\right)$
 6:         $W_u(t+1) \leftarrow W_u(t) - \frac{x}{\alpha}$
 7:         $W_v(t+1) \leftarrow W_v(t) + \frac{(1-\beta)x}{\alpha}$
 8:     else
 9:         $x \leftarrow max\left(0, min(\frac{W_v(t)-W_u(t)}{2-\beta}, \frac{upper-W_u(t)}{1-\beta}, W_v(t)-lower)\right)$
10:         $W_u(t+1) \leftarrow W_u(t) + \frac{(1-\beta)x}{\alpha}$
11:         $W_v(t+1) \leftarrow W_v(t) - \frac{x}{\alpha}$
12:     end if
13: end if
```

As we mentioned above, energy is not transmitted before the average amount of energy in the initial situation is estimated (i.e., $num_{final} > 1$), and, in this phase, energy is actually transmitted and received. The average amount of the energy in the initial situation is stored in W_{ave}. Based on this estimated value, we estimate the upper and lower limits of the final value when the energy balancing is done. Let U and L be positive constant parameters that satisfy $U > L$, let the upper limit be $(1-\beta)W_{ave} * U$ and the lower limit be$(1-\beta)W_{ave} * L$ (Lines 2 to 3). In the simulation experiments of Section 4, appropriate values of U and L are determined from 100 preliminary experiments for each value of α. They meet the conditions that $(1+\epsilon)-$balancing is achieved 99 percent or more and energy loss rate is the smallest. The pattern of energy transfer is classified as follows based on the upper and lower limits. Suppose that agent u and v interact. Without loss of generality, we assume $W_u(t) \geq W_v(t)$.

1. Both of the two agents have energy larger than the upper limit, or have smaller energy than the lower limit:
Energy transmission and reception is not performed. That is, let $x \leftarrow 0$. In this case, energy balancing does not progress.

2. Agent u or v has energy less than the upper limit, and has energy more than the lower limit:
We assume the case where the energy is transmitted and received so that the two agents should have the same amount of energy after the interaction and divide in three cases (a), (b), and (c) according to the amount of energy after the interaction. (a)If the amount of energy is between the upper limit and the lower limit, let $x \leftarrow \frac{W_u(t)-W_v(t)}{2-\beta}$. (b)If the amount of energy is larger than the upper limit, the amount of energy of agent v after the interaction is set to the upper limit, that is, let $x \leftarrow \frac{upper-W_v(t)}{1-\beta}$. (c)If the amount of energy is smaller than the lower limit, the amount of

energy of agent u after the interaction is set to the lower limit, that is, let $x \leftarrow W_u(t) - lower$. These are summarized as the expression $x \leftarrow max\left(0, min(\frac{W_u(t)-W_v(t)}{2-\beta}, \frac{upper-W_v(t)}{1-\beta}, W_u(t) - lower)\right)$. We take a similar approach as P_{BLA} to decide actual amount of transmission energy with parameter α. The transmission amount is a value obtained by dividing x by α ($\alpha \geq 1$). That is, when agent u interacts with agent v at time t ($W_u(t) \geq W_v(t)$), u transmits the following energy amount x' to v.

$$\begin{aligned} x' &= \frac{x}{\alpha} \\ &= max\left(0, min(\frac{W_u(t)-W_v(t)}{2-\beta}, \frac{upper-W_v(t)}{1-\beta}, W_u(t) - lower)\right)/\alpha \end{aligned}$$

3.3 Protocol Based-On-Max-Min P_{BMM}

This protocol P_{BMM} first estimates the maximum and the minimum amounts of energy an agent has in the population. Based on this estimation, energy transfer is performed only if agents with the maximum energy and with the minimum energy interact, so this protocol works only on complete graphs. The idea of this protocol is to avoid wasteful energy transfer as well as P_{BGA}. This protocol restricts cases of energy transfer more severely than P_{BGA} beacause the maximum (resp., minimum) energy in the population is larger (resp., smaller) than the upper (resp., lower) limit of the finally balanced energy of agents.

Each agent has the following variables.
· W_{max}: Estimated maximum energy in the population (Initial value is 100)
· W_{min}: Estimated minimum energy in the population (Initial value is 0)
· t_{max}: Timer to judge whether value of W_{max} is correct or not(Initial value is t_{init} which is a parameter of P_{BMM})
· t_{min}: Timer to judge whether value of W_{min} is correct or not(Initial value is t_{init} which is a parameter of P_{BMM})
· $flag$: Showing the value of W_{max} or W_{min} was updated right before (Initial value is 0)

The pseudocode of P_{BMM} is shown in Protocol 3.

Protocol 3 P_{BMM}

```
$EstimateMax()$
$EstimateMin()$
$TransmitEnergy()$
```

This protocol consists of three functions $EstimateMax()$, $EstimateMin()$,and $TransmitEnergy()$ as shown in Protocol 3, and every agent executes them in this order, that is, the protocol consists of the following three phases:
- Maximum estimation Phase
- Minimum estimation Phase
- Transfer Phase

At each interaction, all phases are performed in this order. However, in Transfer Phase, energy transfer is not performed if estimated values are not calculated.

In Maximum estimation Phase, which is described in function $EstimateMax()$, the maximum amount of energy that an agent has in the population is estimated. The pseudocode of $EstimateMax()$ is shown below.

$EstimateMax()$

```
//agents $u$ and $v$ interact with $u$ as an initiator and $v$ as a responder
```

```
 2: if $W_{max_u} > W_{max_v}$ then
 3:     $W_{max_v} \leftarrow W_{max_u}$
 4:     if $W_{max_u} = W_u(t)$ then
 5:         $t_{max_v} \leftarrow t_{max_u}$
 6:     else
 7:         $t_{max_u} \leftarrow t_{max_u} - 1$
 8:         $t_{max_v} \leftarrow t_{max_u}$
 9:     end if
10: else if $W_{max_u} < W_{max_v}$ then
11:     $W_{max_u} \leftarrow W_{max_v}$
12:     if $W_{max_v} = W_v(t)$ then
13:         $t_{max_u} \leftarrow t_{max_v}$
14:     else
15:         $t_{max_v} \leftarrow t_{max_v} - 1$
16:         $t_{max_u} \leftarrow t_{max_v}$
17:     end if
18: else if $W_{max_u} = W_{max_v}$ then
19:     if $(W_{max_u} = W_u(t)) \vee (W_{max_v} = W_v(t))$ then
20:         $t_{max_u} \leftarrow max(t_{max_u}, t_{max_v})$
21:         $t_{max_v} \leftarrow max(t_{max_u}, t_{max_v})$
22:     else
23:         $t_{max_u} \leftarrow max(t_{max_u}, t_{max_v}) - 1$
24:         $t_{max_v} \leftarrow max(t_{max_u}, t_{max_v}) - 1$
25:     end if
26: end if
27: //update maximum value of energy
28: if $t_{max_u} = t_{max_v} = 0$ then
29:     $W_{max_u} \leftarrow max(W_u(t), W_v(t))$
30:     $W_{max_v} \leftarrow max(W_u(t), W_v(t))$
31:     $t_{max_u} \leftarrow t_{init}$
32:     $t_{max_v} \leftarrow t_{init}$
33:     $flag_u \leftarrow 0$
34:     $flag_v \leftarrow 0$
35: end if
```

Initially, W_{max} of each agent is set to 100 which is the maximum amount of energy each agent can have (or the energy capacity of each agent) and t_{max} of each agent is set to an initial value t_{init} which is determined by preliminary experiments. In the simulation experiments of Section 4, the initial value t_{init} of timer t_{max} is determined from 100 preliminary expriments so that the maximum energy amount is correctly determined in most cases (more than 90 percent of cases). So, incorrect agents that do not have the maximum amount of energy may exchange energy infrequently, but $(1+\epsilon)-$balancing will be achieved eventually. At an interaction, if the larger value of W_{max} two agents have does not equal to the larger energy amount of the two agents, t_{max} is decreased by 1 ,because the estimated value W_{max} may not be correct. Variable W_{max} is updated to the larger value of the interacting agents when t_{max} becomes 0. Repeating this operation, eventually, the correct value of maximum energy in the population is estimated and broadcasted.

The pseudocode describes actions of two agents at an interaction.
(1) If $W_{max_u} \neq W_{max_v}$ (Lines 2 to 17):
Without loss of generality, we assume that $W_{max_u} \geq W_{max_v}$. Then, W_{max_v} is set to the value of W_{max_u} to broadcast. If W_{max_u} is equal (resp., is not equal) to $W_u(t)$, t_{max_u} remains unchanged (resp., is decreased by 1) and t_{max_u} is shared with v.
(2) If $W_{max_u} = W_{max_v}$ (Lines 18 to 26):

If W_{max_u} (=W_{max_v}) is equal to either $W_u(t)$ or $W_v(t)$, t_{max_u} and t_{max_v} are set to the larger value of them and are not decreased. Otherwise, t_{max_u} and t_{max_v} are set to the larger value of them and are decreased bv 1.
After (1) or (2), if $t_{max_u} = t_{max_v} = 0$ (Lines 28 to 35):
Since W_{max_u} and W_{max_v} are considered to be wrong, W_{max_u} and W_{max_v} are updated to the larger amount of energy of the interacting agents and t_{max_u} and t_{max_v} are initialized. $flag_u$ and $flag_v$ are set to 0 to avoid cases where agents that don't have the maximum or minimum energy in the population exchange energy.

In Minimum estimation Phase, which is described in function $EstimateMin()$, the minimum amount of energy that an agent has in the population is estimated. The pseudocode of $EstimateMin()$ is shown below.

$EstimateMin()$

```
1: //agents $u$ and $v$ interact with $u$ as an initiator and $v$ as a responder.
2: **if** $W_{min_v} > W_{min_u}$ **then**
3:     $W_{min_v} \leftarrow W_{min_u}$
4:     **if** $W_{min_u} = W_u(t)$ **then**
5:         $t_{min_v} \leftarrow t_{min_u}$
6:     **else**
7:         $t_{min_u} \leftarrow t_{min_u} - 1$
8:         $t_{min_v} \leftarrow t_{min_u}$
9:     **end if**
10: **else if** $W_{min_v} < W_{min_u}$ **then**
11:     $W_{min_u} \leftarrow W_{min_v}$
12:     **if** $W_{min_v} = W_v(t)$ **then**
13:         $t_{min_u} \leftarrow t_{min_v}$
14:     **else**
15:         $t_{min_v} \leftarrow t_{min_v} - 1$
16:         $t_{min_u} \leftarrow t_{min_v}$
17:     **end if**
18: **else if** $W_{min_u} = W_{min_v}$ **then**
19:     **if** $(W_{min_u} = W_u(t)) \vee (W_{min_v} = W_v(t))$ **then**
20:         $t_{min_u} \leftarrow max(t_{min_u}, t_{min_v})$
21:         $t_{min_v} \leftarrow max(t_{min_u}, t_{min_v})$
22:     **else**
23:         $t_{min_u} \leftarrow max(t_{min_u}, t_{min_v}) - 1$
24:         $t_{min_v} \leftarrow max(t_{min_u}, t_{min_v}) - 1$
25:     **end if**
26: **end if**
27: //update minimum value of energy
28: **if** $t_{min_u} = t_{min_v} = 0$ **then**
29:     $W_{min_u} \leftarrow min(W_u(t), W_v(t))$
30:     $W_{min_v} \leftarrow min(W_u(t), W_v(t))$
31:     $t_{min_u} \leftarrow t_{init}$
32:     $t_{min_v} \leftarrow t_{init}$
33:     $flag_u \leftarrow 0$
34:     $flag_v \leftarrow 0$
35: **end if**
```

Initially, W_{min} each agent has is set to 0 and t_{min} each agent has is set to the initial value which is determined by preliminary experiments. The idea of $EstimateMin()$ is completely same as that of $EstimateMax()$.

In Transfer Phase, which is described in function $TransmitEnergy()$, agents transmit and receive energy, when and only when two agents which have the maximum and minimum energy in the population interacts. The pseudocode of this phase is shown below.

$TransmitEnergy()$

```
1: if $(W_{max_u} = W_u(t)) \wedge (W_{min_v} = W_v(t))$ then
2:     if $flag_u \cdot flag_v > 0$ then
3:         $x \leftarrow \frac{W_u(t)-W_v(t)}{\alpha(2-\beta)}$
4:         $W_u(t+1) \leftarrow W_u(t) - x$
5:         $W_v(t+1) \leftarrow W_v(t) + (1-\beta)x$
6:     end if
7:     $flag_u \leftarrow 1$
8:     $flag_v \leftarrow 1$
9: else if $(W_{max_v} = W_v(t)) \wedge (W_{min_u} = W_u(t))$ then
10:    if $flag_u \cdot flag_v > 0$ then
11:        $x \leftarrow \frac{W_v(t)-W_u(t)}{\alpha(2-\beta)}$
12:        $W_u(t+1) \leftarrow W_u(t) + (1-\beta)x$
13:        $W_v(t+1) \leftarrow W_v(t) - x$
14:    end if
15:    $flag_u \leftarrow 1$
16:    $flag_v \leftarrow 1$
17: end if
```

In this phase, only two agents which have the maximum and minimum energy in the population exchange energy. Without loss of generality, we assume that $(W_{max_u} = W_u(t)) \wedge (W_{min_v} = W_v(t))$ (Lines 1 to 8). Agents that upadated W_{max} or W_{min} right before may exchange energy even if they do not have the maximum or minimum energy in the population. $flag$ prevents this case from happening. If $flag$ is 0, energy transfer is not performed and $flag$ is set to 1 to perform energy transfer next time. Energy transmisson amount is similar to that of P_{BLA}.

4 Simulation Experiment

In this section, we evaluate three algorithms P_{BLA}, P_{BGA}, and P_{BMM} by simulation experiments using a C program. The network we consider are complete graphs and grid graphs. The metrics to evaluate protocols are the energy loss rate and the execution time.

4.1 Simulation Settings

In this subsection, we explain the settings in the simulation experiment. In complete graphs, all the ordered pairs of agents are in E. That is, for each distinct agents u and v, the probability that u is an initiator and v is a responder is $\frac{1}{n(n-1)}$. Complete graphs are most commonly used on population protocol model. In grid graphs, the ordered pairs of agents that are next to each other up, down, left or right are in E. Grid graphs are commonly used on sensor network models. The grid graphs are simulated only when the number of agents is 100 (10 × 10). Each agent is assigned an integer value drawn from 0 to 99 uniformly at random as the energy amount in the initial situation. That is, $Pr(W_v(0) = i) = \frac{1}{100}$ holds for any $v \in V$ and $i = 0, 1, \ldots, 99$. In each experiment, an execution of a protocol is simulated until an energy distribution of $1.01 - balancing$ (i.e. $\epsilon = 0.01$) is obtained. We simulate 100 executions for each setting and evaluate the energy loss rate and execution time of the executions in average. Regarding the environment, we simulate three different environment with loss rate $\beta = 0.2, 0.4$ and 0.6. Table 1 shows the set values of various parameters used in the experiment.

Table 1: the set values of various parameters

parameter	**value**
Number of agents n (complete graph)	$50, 100, 200$
Number of agents n (grid graph)	100
Initial Energy	from 0 to 100
Loss rate β	$0.2, 0.4, 0.6$
ϵ	0.01

4.2 Results and Discussion

Protocols are compared from two aspects, the energy loss rate and the execution time. We evaluate P_{BLA}, P_{BGA}, and P_{BMM} by changing the parameter of α of energy transmission fraction.

By preliminary experiment for P_{BGA}, we obtain the optimal values for the upper and lower limits of balanced energy of agents when the energy balancing process is achieved for each value of α. Among the combinations of value of U and L when the upper limit is set to $(1-\beta)W_{ave}*U$ and the lower limit is set to $(1-\beta)W_{ave}*L$, we choose the optimum values, which achieve $(1+\epsilon)-balancing$ with 99 percent or more in 100 experiments and the energy loss rate is the lowest. This combinations are calculated by simulation with three significant figures for $\beta = 0.2, 0.4$ and 0.6 and, values of U and L are set to the values for simulation.

4.2.1 Complete Graph

The simulation results on the complete graphs are shown below.

Figure 1 shows the results of the energy loss rate of three protocols for 100 agents.

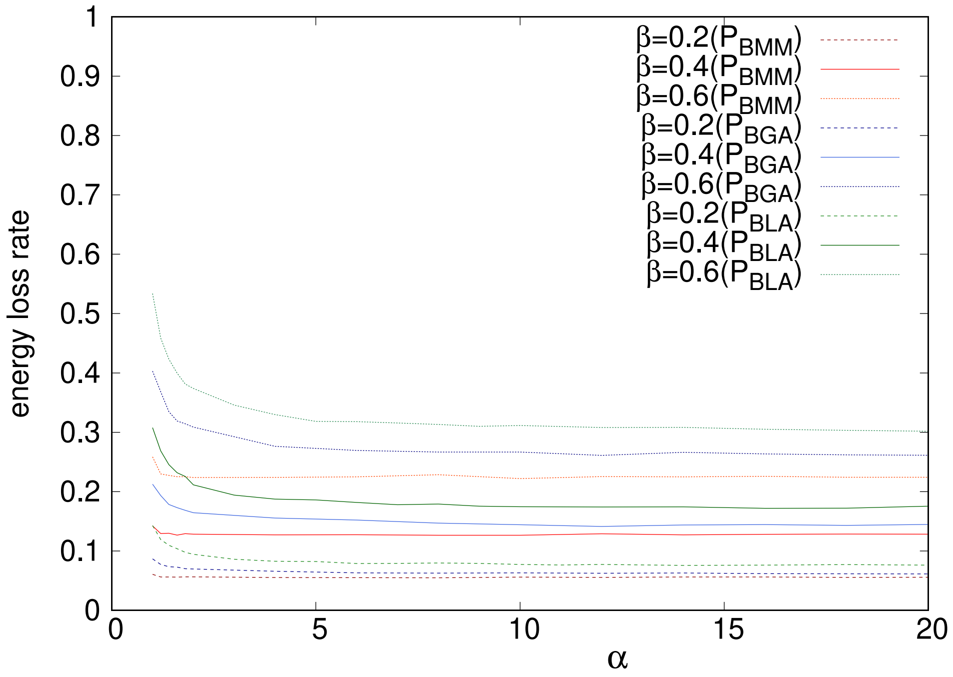

Figure 1: Comparison of energy loss rate ($\#agents = 100$, complete graph)

As anticipated in Section 3.1, the energy loss rate decreases as the value of α increases in any protocol. However, it is confirmed that the energy loss rate does not continue to decrease but converges to a certain value. Also, as value of β increases, the energy loss per interaction increases, so the energy loss rate also increases. The timing at which the energy loss rate converges is almost the same for any value of β of any protocol. The energy loss rate hardly changes in the range of $\alpha > 5$. The comparison result shows that for any value of β, the energy loss rate of P_{BMM} is the lowest, by that of P_{BGA} and then, that of P_{BLA}. This implies that P_{BGA} effectively uses the estimated global information (or the average of the initial amount of energy) to reduce energy loss due to wasteful energy transfer, and P_{BMM} further effectively uses the estimated global information (or agents which have the maximum or minimum energy in the population).

The similar tendency for the energy loss rate is observed also in the case where the number of agents are 50 and 200. In addition, in the case where the number of agents are 50 and 200, the energy loss rate is not significantly changed.

Next, Figures 2 and 3 show the result of comparing the execution time of protocols for 100 agents.

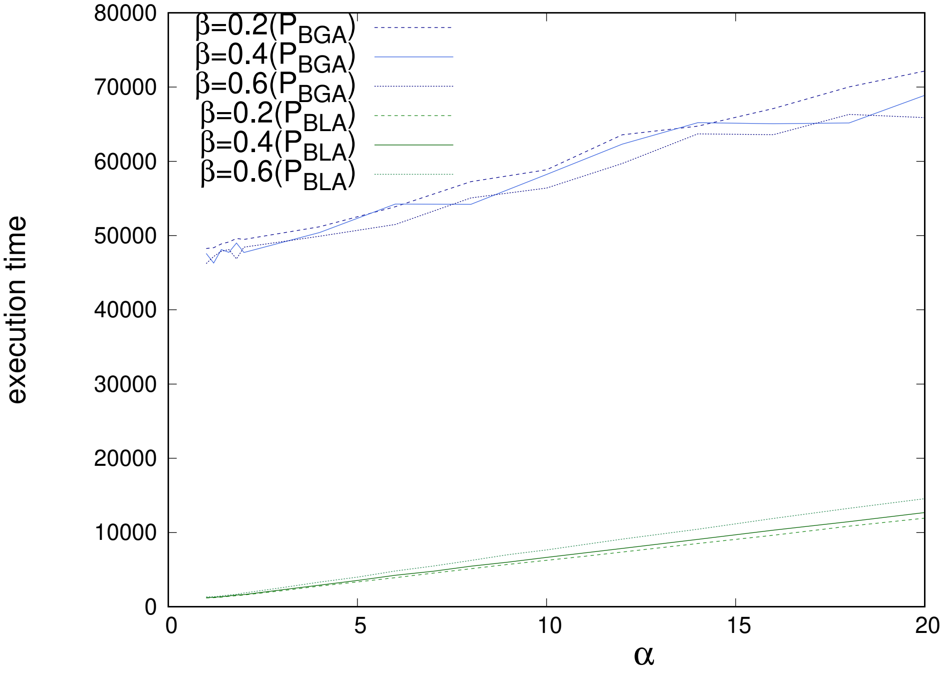

Figure 2: Comparison of execution time ($\#agents = 100$, complete graph)

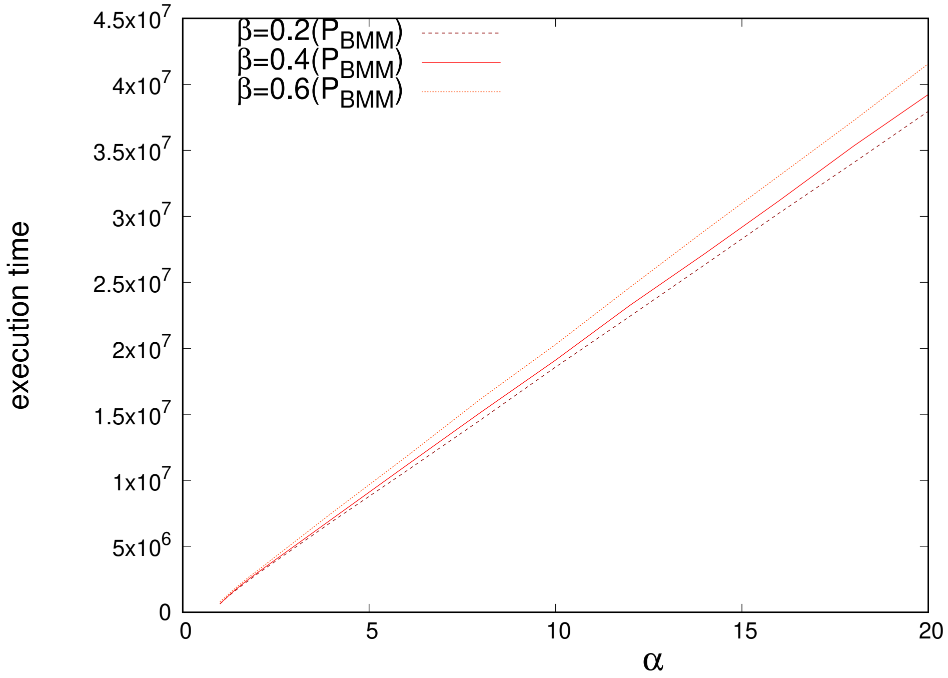

Figure 3: Comparison of execution time ($\#agents = 100$, complete graph)

In any protocols, the execution time increases as value of α increases for any value of β. This is because the amount of transmitted energy per one interaction are divided by α. P_{BLA} and P_{BMM} has larger execution time as the value of β is larger. As for P_{BGA}, the execution time tends to decrease as the value of β increases. The comparison result shows that for any value of β, the execution time of P_{BMM} is the longest, followed by P_{BGA} and then P_{BLA}. The reason why the execution time of P_{BMM} is much longer than that of P_{BGA} or P_{BLA} is that P_{BMM} takes a very long time to estimate agents that have the maximum or minimum energy in the population and P_{BMM} has very strict restriction to interact. The reason why the execution time of P_{BGA} is longer than that of P_{BLA} is that P_{BGA} takes a long time to judge whether the only one leader has been elected or not. The similar tendency for execution time is observed even when the number of agents are 50 and 200.

By comparing the protocols in the energy loss rate and the execution time in total, for any value of β, the energy loss rate of P_{BMM} is the best (or smallest) followed in order by that of P_{BGA} and that of P_{BLA} and the execution time of P_{BLA} is the best (or shortest) followed in order by that of P_{BGA} and that of P_{BMM} for arbitrary value of α. The same result is obtained when the number of agents are 100 and 200.

4.2.2 Grid Graph

In the grid graph as well, we compare the proposed protocols for the energy loss rate and the execution time. P_{BMM} only assumes complete graphs, so only P_{BLA} and P_{BGA} are compared. Since it is found that the similar tendency is shown when changing the number of agents in the complete graph, we simulate with grid graph only with 100 agents.

Figure 4 shows the result of the energy loss rate of protocols for 100 agents.

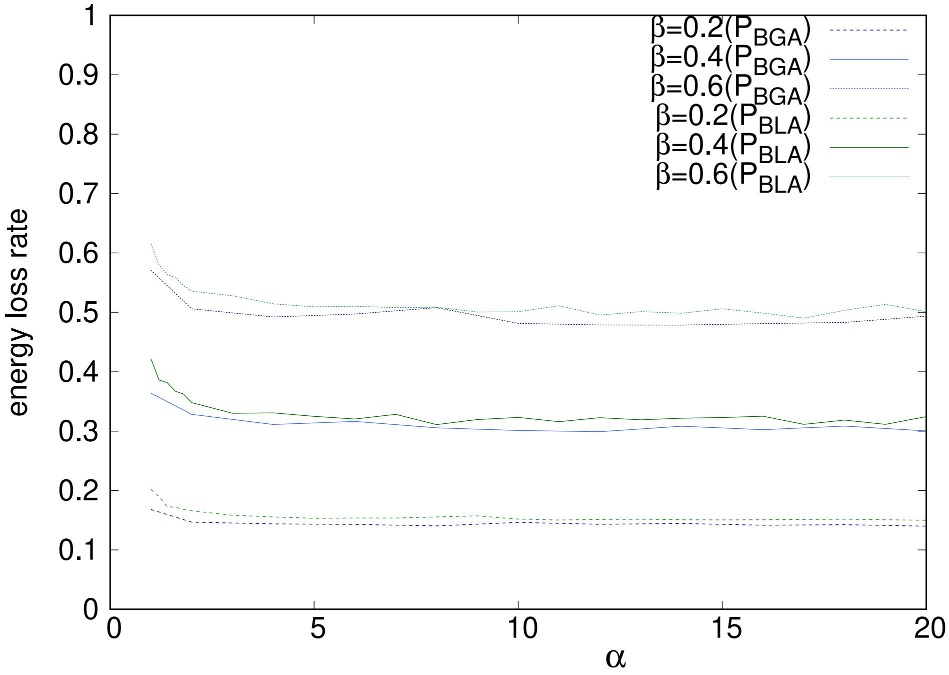

Figure 4: Comparison of energy loss rate ($\#agents = 100$, grid graph)

As with the complete graphs, the energy loss rate of P_{BGA} is smaller than that of P_{BLA} for any value of β.

Next, Figure 5 shows the result of comparing the execution time of 100 agents.

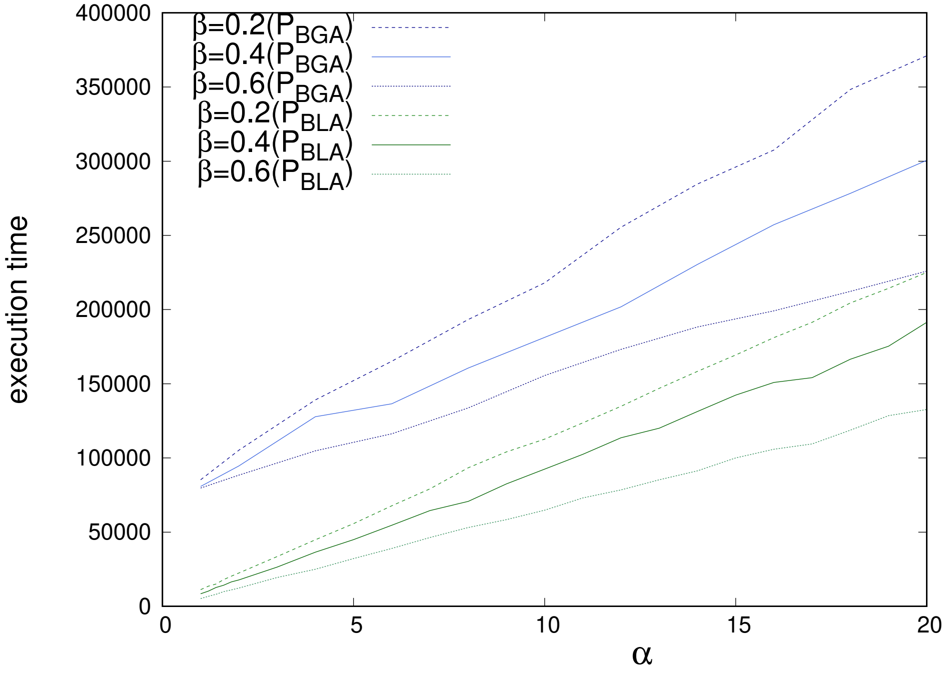

Figure 5: Comparison of execution time ($\#agents = 100$, grid graph)

As with the complete graphs, the execution time of P_{BGA} is longer than that of P_{BLA} for any value of β.

By comparing the protocols in the energy loss rate and the execution time in total, as with the complete graphs, for any value of β, the energy loss rate of P_{BGA} is better than that of P_{BLA} while the execution time of P_{BLA} is better than that of P_{BLA} for arbitrary value of α.

4.3 Summary of resuls

In summary, it turns out, as for energy loss rate, P_{BMM} (a protocol which globally estimates agents that have maximum or minimum energy in the population) is the best followed in order by P_{BGA} (a protocol which globally estimates average of initial energy) and P_{BLA} (a protocol which estimates the energy transmission amount depending only on the interacting two agents). On the other hand, as for execution time, P_{BMM} is the worst followed in order by P_{BGA} and P_{BLA}.

5 Conclusion

In this paper, we proposed three protocols for balancing energy of agents in sensor networks and evaluate the protocols in energy loss rate and execution time by simulation experiment. We showed that the energy loss rate of the sophisticated protocol P_{BMM} and P_{BGA} are smaller than that of the simple protocol P_{BLA}, but the execution time of P_{BMM} and P_{BGA} are longer than that of P_{BLA}. Further improvement of energy loss rate and execution time are our future work. Also, we will consider how much the energy loss rate and execution time of the proposed protocols are inferior to the theoretical optimum value. Furthermore, we will evaluate the case where the distribution probability of initial energy is not uniform.

Acknowledgements

This work is supported in part by JSPS Grant-in-Aid for Scientific Research 17K19977, 19K11826, 18K18000, and JST SICORP.

References

[1] Dana Angluin, James Aspnes, Zoe Diamadi, Michael J. Fischer, and Rene Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, pages 235–253, March 2006.

[2] IanF.Akyildiz, WeilianSu Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, pages 102–114, August 2002.

[3] Jun Luo and Ying He. Geoquorum: Load balancing and energy efficient data access in wireless sensor networks. *in Proceedings of 30th IEEE Conference on Computer Communications (INFOCOM)*, pages 616–620, 2011.

[4] Adelina Madhja, Sotiris Nikoletseas, Theofanis P. Raptis, Christoforos Raptopoulos, and Dimitrios Tsolovos. Peer-to-peer wireless energy transfer in populations of very weak mobile nodes. *in Proceedings of Wireless Communications and Networking Conference Workshop (WCNCW)*, pages 1–6, 2017.

[5] Adelina Madhja, Sotiris Nikoletseas, Dimitrios Tsolovos, and Alexandros A. Voudouris. Peer-to-peer energy-aware tree network formation. *Proceedings of the 16th ACM International Symposium on Mobility Management and Wireless Access (MOBIWAC)*, pages 1–8, 2018.

[6] Sotiris Nikoletseas, Theofanis P. Raptis, and Christoforos Raptopoulos. Interactive wireless charging for weighted energy balance. *in Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS)*, page 119, 2016.

[7] Sotiris Nikoletseas, Yuanyuan Yang, and ApostolosGeorgiadis. *Wireless power transfer algorithms, technologies and applications in ad hoc communication networks.* Springer International Publishing, July 2016.

[8] Jose Rolim. Energy balance mechanisms and lifetime optimization of wireless networks. *in Proceedings of International Conference on Contemporary Computing (IC3)*, page 1, 2011.

[9] Yuichi Sudo, Fukuhito Ooshita, Hirotsugu Kakugawa, Toshimitsu Masuzawa, Ajoy K. Datta, and Lawrence L. Larmore. Loosely-stabilizing leader election with polylogarithmic convergence time. *Proceedings of 22nd International Conference on Principles of Distributed Systems (OPODIS)*, pages 301–316, 2018.