Master's Thesis

Title

Spreading Factor Allocation Method Adaptive to Changing Environments for LoRaWAN Based on Thermodynamical Genetic Algorithm

Supervisor

Professor Masayuki Murata

Author

Yuki Fujita

February 3rd, 2022

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Spreading Factor Allocation Method Adaptive to Changing Environments for LoRaWAN Based on Thermodynamical Genetic Algorithm

Yuki Fujita

Abstract

LPWA is a general term for low-power and wide-area communication technologies that are expected to be used as networks for IoT. In particular, LoRaWAN has become a major research target in LPWA because of its ease of development and the possibility of building self-managed networks. In LoRaWAN, the data rate of each node can be dynamically controlled by the gateway node through changing the scaling factor parameter of nodes, and this control can be performed according to the network conditions. However, it is difficult to grasp the individual states of a large number of nodes immediately, calculate the optimal data rate, and assign the appropriate scaling factor to the nodes due to the low communication speed of LoRaWAN compared to conventional wireless networks. In this thesis, we propose a spreading factor allocation method for LoRaWAN that simultaneously improves throughput and network lifetime even in an environment where network conditions fluctuate. Our proposed method is based on the evolutionary mechanism of living organisms. Specifically, in contrast to the traditional genetic algorithm, which is one of the evolutionary algorithms, we use the thermodynamical genetic algorithm, which can adapt to dynamically changing environments by maintaining the diversity of the population. The thermodynamical genetic algorithm balances the fitness and diversity of each individual. The diversity of individuals is expected to contribute greatly to the acquisition of better solutions. In addition, for environments where the optimal solution differs greatly before and after environmental changes, we use a feedback mechanism for thermodynamical genetic algorithm to keep the diversity of the population at a predetermined value. Through computer simulation, we show that our proposed method can perform appropriate control of the scaling factor adaptive to dynamically changing network environments. Compared to a conventional genetic algorithm that do not maintain the diversity of population, our proposal can achieve higher fitness in fluctuating environments.

Keywords

LPWA

Evolutionary algorithm

TDGA

Diversity

Multi-objective optimization

Contents

List of Figures

List of Tables

1 Introduction

Low Power Wide Area (LPWA) is one of the wireless communication technologies with low power consumption and long range capability, and it is already widely used in Europe as a component of the Internet of Things (IoT) [1]. A Long Range Wide Area Network (LoRaWAN) is one of the communication networks belonging to LPWA, which consists of nodes and gateways that use LoRa modulation [2]. Generally the communication speed of LPWA techniques is slow, ranging from a few hundred bps to a few tens of kbps, it provides power efficiency better than Bluetooth and ZigBee, and communication distance as well as or longer than 3G and LTE mobile communication. There are many other LPWA standards such as Sigfox and NB-IoT in addition to LoRaWAN, but compared to other standards LoRaWAN has the feature of being easy to develop and build user's own networks, and therefore it has become a major research target for IoT networks [3].

Since LoRa uses unlicensed bands and its communication modules are readily available, users can freely construct their own networks. In the future, there will be a situation where multiple self-managed networks with many LoRa nodes exist in the same area, and therefore wireless communication will cause packet collisions. LoRa uses the ALOHA protocol in its MAC layer. Therefore, since the data rate of LoRa communication is low, the data transmission time of the node becomes long, which increases the probability of packet collisions.

Although the LoRa chip has a carrier sense function, the antenna reception sensitivity of LoRa is higher than existing wireless communication modules. If using the Clear Channel Assessment (CCA) threshold of about -80dB used in IEEE 802.11, since the signal lower than the threshold reach the gateway, collisions are expected to occur. On the other hand, if the threshold is lower, nodes are exposed to more wireless communications from more nodes due to the very wide communication range of LoRa, and the possibility that the wireless channel is busy is higher.

In the current LoRa communication standard, there are multiple data rates available for nodes to take into account the interference. In LoRaWAN, the gateway can dynamically control the data rate (adaptive data rate; ADR) by changing the Spreading Factor (SF) of the node through the control signal [4]. Note that even in the same frequency band, radio signals with different spreading factors can be received simultaneously by LoRa gateways [5]. In general, the larger the spreading factor, the lower the data rate and the longer the transmission time, which increases the power consumption, while the signal-to-noise ratio (SNR) increases, which increases the reception sensitivity and the communication range.

Previous studies of LoRaWAN showed performance related to coverage and scalability [6, 7]. In recent years, studies have been conducted to improve throughput and extend service lifetime by controlling the data rate [8]. In the original specification of LoRaWAN, ADR is performed by the gateway [4]. The problem of determining the data rate of individual nodes in order to maximize the throughput is a combinatorial optimization problem, and it is difficult to find the optimal solution in a realistic time when there are huge number of LoRa nodes.

In general, wireless congestion and communication quality fluctuate due to various environmental variations. If we apply such an optimization method that takes enough time to collect information from all nodes and solve the optimization problem, performance degradation during this period may be unavoidable. In this paper, we focus on the evolutionary mechanism of organisms that have survived under various environmental changes in order to realize the optimization of ADR in LoRaWAN. Specifically, we use Thermodynamic Genetic Algorithm (TDGA) to determine the spreading factor of individual node in LoRaWAN.

TDGA is based on the genetic algorithm (GA), one of the evolutionary algorithms, and it has a novel gene selection algorithm that provides the diversity of individuals to obtain an adaptability to dynamic environments. TDGA takes into account not only the fitness of each individual but also the diversity of the population when selecting individuals from the population for the next generation, while genetic manipulations such as crossover and mutation are the same as GA. When the temperature parameter of TDGA is set to a low value, the focus is on fitness, and when it is set to a high value, the focus is on diversity.

We have demonstrated the performance of our TDGA-based SF-allocation method by computer simulation assuming a LoRaWAN application that considers node mobility [9] and gateway failures. In particular, we have shown that our proposal can find better solutions faster than a GA-based method when the degree of congestion changes dynamically.

Moreover, since TDGA has a characteristics that the diversity of population decreases when individuals have better fitness, we apply a feedback mechanism to TDGA to keep the diversity of individuals constant. This feedback TDGA is called FTDGA [10].

This paper is organized as follows. First, in Section 2, we describe recent related researches on LoRaWAN. Next, Section 3 describes the thermodynamic genetic algorithm used in this paper. Section 4 describes the communication model of LoRaWAN and the proposed method using TDGA, and in Section 5 we evaluate the effectiveness of the proposed method. Finally, we conclude and discuss future work in Section 6.

2 Related work

In this section, we describe related work on spreading factor control in LoRaWAN and discuss the difference from our proposal.

Reference [11] finds appropriate parameter values using reinforcement learning. Specifically, they use a deep learning approach to tackle the distribution problem for optimizing network resources such as spreading factor, transmission power, and channel allocation in LoRaWAN. In [12], the authors propose a method to improve performance by spreading factor allocation in LoRaWAN. They define an optimization problem of spreading factor allocation to maximize the data reception rate under the constraint on the average power consumption of all nodes. By solving this problem, it is possible to improve the performance under the average power consumption constraint of all nodes. This paper further develops a meta-heuristic method, which is a method for solving the defined problem based on distributed genetic algorithm. Reference [13] investigates the impact of scalability and densification of nodes and gateways on the reliability of the system, taking into account the capture effect. An optimization problem is proposed to derive the node distribution in LoRaWAN networks with multiple gateways at different spreading factor. They also introduce an adaptive algorithm that can easily optimize the spreading factor by adjusting the threshold of the signal-to-noise ratio. Reference [14] proposes a spreading factor allocation approach that pays attention to the traffic load on both the spreading factor and the channel. In [15], an analysis is performed to improve the average system packet success probability (PSP) of LoRa system in ALOHA random access protocol. As a result, they derived a lower bound for the average system PSP. They also showed that the average system PSP can be maximized by properly assigning a spreading factor to each traffic, which also maximizes the node connectivity. In [16], the Greedy method is used for the fairness of power consumption as a objective function. Reference [17] shows that spreading factor assignment using the k-means method improves the success rate of communication.

These previous work have optimized the various parameters of LPWA networks. However they do not take into account environmental changes such as the movement of LoRa nodes in the actual operation of LoRaWAN, and do not show whether it is possible to quickly follow environmental changes. One of the goal of our research is to find better solutions quickly in a fluctuating environment, and to do so we propose a method to keep the diversity of solution candidates at a high value.

3 Thermodynamical Genetic Algorithm

3.1 Genetic algorithm

Genetic Algorithm (GA) is a meta-heuristic algorithm that applies the process of biological evolution to solve optimization problems. In genetic algorithms, data sequence is mapped to elements of the problem to be solved as genomes. A set of individuals represented by the genome is called *population.* An approximate solution to the optimization problem is obtained by performing genetic operations on the iteration (generation) to evolve it.

A typical GA is performed in the following steps. Let the size of the population be N and the number of iterations be G.

0. Prepare a null set of the current generation and a null set of the next generation, respectively. Generate N individuals with random genes for the first generation. Add These individuals into the current generation.
1. Cross two randomly selected individuals from the set of the current generation and add it to the the next generation set.
2. Repeat 1. until the number of individuals in the next generation set becomes N.
3. Add a copy of each individual in the set of the current generation to the set of the next generation.
4. Mutate at a constant rate for each individual in the set of the next generations.
5. Select N individuals from the set of the next generation and add them to the current generation.
6. Return to 1. If the process has been repeated G times, it will output the individual that maximizes the evaluation function (fitness function) from the set of the next generation and finish.

The genetic operations that appear during each step are as follows.

Crossover Crossover corresponds to mating in living organisms, in which two new offspring are produced using the genomes of the two parental individuals (Step. 1). In this paper, we use a crossover method called uniform crossover. In uniform crossover, each gene of a parent is replaced with a gene of the same locus as the other parent with a certain probability, and these two genomes are used as children.

Mutation Mutation corresponds to the mutation of a gene in an organism (Step. 4). In this paper, for each gene, the probability of mutation is changed to a spreading factor different from the current value with a certain probability.

Selection Selection corresponds to the natural selection of organisms, where the degree of adaptation of each individual is calculated based on an evaluation function, and the more adaptable the individual is, the more likely it is to be selected (Step. 5). In this paper, we use roulette selection as a method of selection. In roulette selection, the probability p_i that each individual i of generation P is selected is determined by equation (1), and a total of N individuals are selected with this probability.

$$p_i = \frac{f(i)}{\sum\limits_{k \in P} f(k)} \tag{1}$$

f is the fitness function defined in section 4.3.

In contrast to the TDGA described below, we will call this simple genetic algorithm SGA (Simple GA).

3.2 TDGA: Thermodynamical genetic algorithm

It is known that the state of a system in thermal equilibrium at temperature T is a probability distribution that minimizes the free energy F in Eq. (2) (free energy minimization principle).

$$F = \langle E \rangle - TH \tag{2}$$

$\langle E \rangle$ is the average internal energy of the system and H is the entropy. When the free energy is minimized, the right-hand side of Eq. (2) can be interpreted as $\langle E \rangle$ is the term for energy minimization and $-H$ is the term for pursuing state diversity, and T is treated as a parameter that harmonizes the two. In TDGA, the mechanism of deriving $\langle E \rangle$ and

H in such a way as to minimize the free energy F is applied to GA. In TDGA, individuals with a high degree of fitness are retained in the solution space of the target problem, while those with a high degree of difference from other individuals are actively retained in the next generation. This can be expected to function effectively in discovering new solutions after the objective function has changed.

The specific operation of TDGA is as follows. The number of individuals N, the maximum number of generations G, and the temperature schedule function $T(t)$ are determined in advance.

0. Set the number of generations $t = 0$. Initialize the population $P(0)$ with random individuals.

1. Let e be the elite individual with the greatest fitness.

2. Generate N childrens from $P(t)$ by crossover.

3. Construct a candidate population $P'(t)$ of the next generation from $P(t)$ and the childrens generated in Step 2. And mutate $P'(t)$.

4. Add e to the next generation population $P(t + 1)$.

5. Set the number of individuals selected as the next generation of individuals $i = 1$.

6. Select one individual from $P'(t)$ and move it to the next generation population $P(t + 1)$. In this operation, assuming that the i-th individual h from $P'(t)$ is added to $P(t+1)$, select one individual h which minimizes $P(t+1)$'s free energy F (Eq. (3)) as the i-th individual.

7. $i = i + 1$, and if $i < N$, go back to Step 6.

8. $i = i + 1$, and if $t < G$, go back to Step 1.

$$
\begin{aligned}
F &= \langle E \rangle - T(t)H \\
&= \frac{\sum_{l=1}^{i} E_l(P(t+1))}{i} - T(t)H
\end{aligned}
\tag{3}
$$

In Eq. (3), TDGA expresses the average energy $\langle E \rangle$ of the system as the negative of the fitness of each individual. The simplest form of entropy H, which represents the diversity of a system, is H^{ALL}, which treats all loci together.

$$H^{ALL} = -\sum_i p_x \log p_x \tag{4}$$

p_x is the probability of the existence of species x in the population. However, in GA, the size of the population is very small compared to the total number of possible species, and therefore even if the population consists of only a few different individuals, H^{ALL} will be large and impractical [18]. Therefore, TDGA uses alternative entropy definitions; H^1 per locus and H^2 per two neighboring loci [19]. In this report, we use H^1. H^1 is defined by the following equation.

$$H^1 = \sum_{k=1}^{M} H_k^1, \quad H_k^1 = -\sum_{\alpha \in \text{allele}} P_\alpha^k \log P_\alpha^k \tag{5}$$

In Eq. (5), H_k^1 represents the entropy for the gene at locus k of the population, and P_α^k is the probability of the existence of allele α at locus k.

3.3 FTDGA: Feedback TDGA

In the original TDGA, temperature T is a parameter to be determined in advance, and considering the minimization of the free energy F in Eq. (3), the entropy of the population decreases as the fitness value of the candidate population increases with each step of the solution search. Feedback TDGA (FTDGA) addresses this problem by dynamically changing the T in response to changes in the environment, allowing it to cope with environmental changes of unexpected scales.

To realize a feedback type control, a temperature scheduling function $\mathcal{T}(t)$ defined as Eq. (6) is proposed in [10]. This control preserves the search capability of the TDGA by keeping the entropy at the target entropy H^*.

$$\mathcal{T}(t) = \exp(\tau(H^* - H))\mathcal{T}(t-1) \tag{6}$$

where τ is a parameter called feedback gain.

4 Spreading factor allocation method based on TDGA

4.1 LoRaWAN communication model

4.1.1 LoRa modulation

In LoRa modulation, the modulation scheme used in LoRaWAN, the bandwidth is divided equally, and the signal is sent starting at an arbitrary frequency and increasing in frequency as time passes. Once the signal has increased to the upper limit of the bandwidth, it is increased from the lower limit to the frequency at the start. This one cycle of the signal is called a symbol, and the data is represented by the frequency at the beginning of the symbol. The number of bits of data assigned to one symbol is called the spreading factor. When the spreading factor is SF, the data that can be expressed in one symbol is from 0 to $2^{SF} - 1$.

Increasing the spreading factor by one doubles the length and the number of bandwidth divisions of a symbol. Therefore, the larger the spreading factor, the more the signal is spread out in time, making it more resistant to noise, but lowering the data rate and increasing the transmission time. If the bandwidth is BW, the data rate is expressed as $\frac{BW}{2^{SF}}SF$. Signals with different spreading factors have different periods of frequency change, so even if they are transmitted simultaneously, the time of frequency overlap is very short. This makes them regarded as practically non-interfering. An example of LoRa modulation is shown in Fig. 1.

In the following, the available bandwidth, spreading rate, and receiver sensitivity for each spreading factor follow the SX1276 module of Semtech, which designs and develops LoRa modules [20].

4.1.2 MAC layer model

In LoRaWAN, there are three communication classes in the MAC layer: Class A, Class B, and Class C. In Class A of LoRaWAN, ALOHA (pure ALOHA) is used as the communication protocol, and each node sends data packets to the gateway at an arbitrary timing. In Class B, in addition to the communication in Class A, the gateway can broadcast data to all nodes at regular intervals. In Class C, the gateway can send data to nodes at any

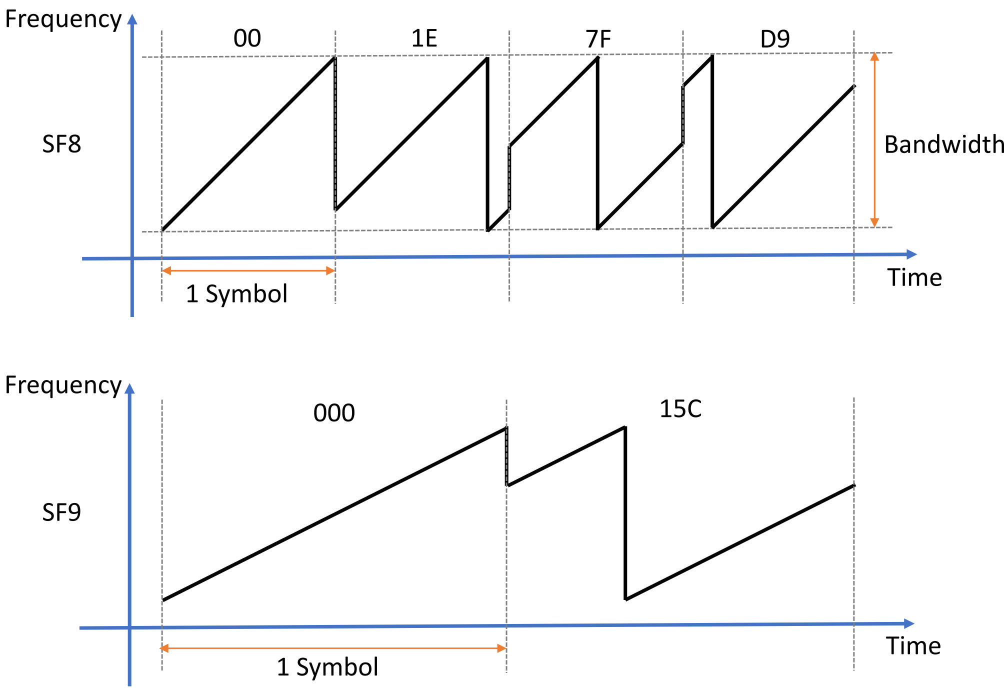

Figure 1: An example of LoRa modulation. The number at the top of each symbol indicates the data that the symbol represents (hexadecimal number).

time. Class C is intended for use with nodes that is always powered. In the method we propose, we assume communication in Class B.

As described in the previous section, when two or more data packets sent with different spreading factors arrive at the gateway at the same time, the frequencies do not overlap, so we assume that all packets can be received normally in this case. If data packets with the same spreading factor are received at the same time, they collide with each other and neither can be received. In this paper, it is assumed that all the packets from the gateway to the node can be successfully transmitted because the signal strength is sufficiently high, and that the communication from the node to the gateway and the communication from the gateway to the node are not affected by each other because they use different channels.

4.1.3 LoRaWAN model

We assume situations where multiple movable LoRa nodes transmit data to multiple gateways. The LoRa nodes periodically generate data and transmit them to the gateway immediately after they are generated. Transmission after data generation is performed only once, no re-transmission is performed. The data sent by a node is considered to have been successfully transmitted if it is successfully received and decoded by at least one gateway.

When two or more signals arrive at a gateway at the same time, if they use the same spreading factor, both signals will fail to be decoded, but if they use different spreading factors, both signals will be successfully decoded. Moreover, it is assumed that there will be attenuation of the signal strength depending on the distance between the LoRa node and the gateway, which may cause probabilistic decoding errors and failure in decoding.

The spreading factor used by the LoRa node is changed by control information sent from the gateway, but sending such information to each node has a very large overhead in LPWA, which has a slow communication speed. Therefore, the area to be observed is divided into several sub-areas, and the LoRa nodes belonging to the same sub-area use the same spreading factor. The control information from the gateway is sent to each sub-area by broadcast at once. At this time, each LoRa node needs to know to which sub-area it currently belongs, and it is assumed that the location information can be obtained by GPS (Global Positioning System). It is also assumed that each LoRa node can always get accurate information about the sub-area to which it belongs because the position estimation error of GPS is very small compared to the size of the sub-area.

4.2 Encoding representation

The number of genes in an individual is the same as the number of sub-areas in the LoRaWAN model, and the spreading factor of each locus and sub-area corresponds as shown in Fig. 2. Therefore, each gene locus will have a value corresponding to available SFs (SF7 to SF12).

4.3 Fitness functions

LoRaWAN applications have various performance objectives, and we aim to improve the data arrival rate and save energy, which are generally important in many applications. In this thesis, we use the sum of F_{arr} and F_{pow}, which will be explained later, as fitness

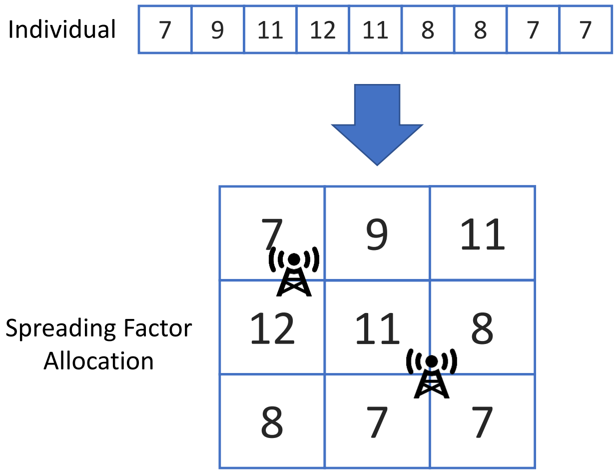

Figure 2: Genes expression and spreading factor

function. The exact solution in this case requires a number of searches proportional to $N_{SF}^{N_{area}}$, where N_{SF} is the number of selectable spreading factors and N_{area} is the number of subareas. For example, if LoRa assigns 6 available spreading factors to 100 sub-areas, $6^{100} \simeq 6.5 \times 10^{78}$ searches are required. In this paper, we use the FTDGA to find a solution to this combinatorial problem.

In the ALOHA protocol, we assume that there are n nodes on the same link, and each node generates an average of p packets per unit time. In this case, the packet generation rate μ from all nodes is equal to np. Therefore, the probability $P(x)$ of x packets being generated in a unit time is represented by the Poisson distribution in Eq. (7).

$$P(x) = \frac{\mu^x}{x!} e^{-\mu} \tag{7}$$

If the transmission time of a single packet is T, then in order for a packet generated at time 0 to be transmitted without collision, no other packets must have been generated between time $-T$ and T. This probability P_{ALOHA} is Eq. (8).

$$P_{ALOHA} = P(0)^2 = e^{-2\mu T} \tag{8}$$

4.3.1 Fitness function for the arrival rate at a single gateway

In the LoRaWAN communication system, each node uses a different spreading factor. As mentioned above, communications between nodes using different spreading factors do not collide each other. In addition, the packet error rate in the transmission path is different for each spreading factor. When the packet error rate in the transmission path is PER, the data arrival rate F_{arr} of nodes in LoRaWAN is Eq. (9).

$$F_{arr} = (1 - PER)e^{-2\mu T} \tag{9}$$

However, in calculating the fitness function, PER is calculated assuming that all nodes are located in the center of the sub-area to which they belong. In addition, n is the sum of the number of nodes using the same spreading rate, and p is the reciprocal of the packet generation period. F_{arr} is calculated for each node, and the average value is used as the fitness for the arrival rate.

4.3.2 Fitness function for the arrival rate at multiple gateways

In LoRaWAN, it is possible to use multiple gateways. In that case, it is assumed that there are multiple gateways in the communication range of a node. In this paper, it is assumed that all gateways are included in the communication range of a node m, and each gateway is called $\mathrm{BS_i}$ ($i = 0 \ldots N_{BS} - 1$, where N_{BS} is the total number of gateways), according to the LoRaWAN specification, the data sent by a node should be successfully received by at least one gateway. Therefore, the arrival rate (F_{arr}) is as follows.

$$F_{arr} = N_{all}^{-1} \sum_{m=0}^{N_{all}} \left\{ \prod_{i=1}^{N_{\mathrm{BS}}} (1 - PER(m,i))(1 - e^{-2\mu T}) \right\} \tag{10}$$

where $PER(m,i)$ is the packet error rate determined according to the distance between node m and gateway i, and N_{all} is the total number of nodes.

4.3.3 Fitness function for the total power consumption of nodes

We use Eq. (11) as the fitness function for power consumption. Here, n is the number of nodes to be controlled, Pow(SF) is the power consumption required for one data transmission at the spreading factor SF, and P is the sum of the power consumption required for one data transmission for all nodes. We use the power consumption for the case of the largest spreading factor of 12 and the smallest of 7, and keep $F_{pow-total}$ in the range of 0 to 1.

$$F_{pow-total} = -\frac{P - n\mathrm{Pow}(12)}{n(\mathrm{Pow}(12) - \mathrm{Pow}(7))} \tag{11}$$

Note that we set a penalty on the power consumption. If the total power consumption of all nodes exceeds the upper limit value, the fitness of the power is set to 1/100 of the original value.

5 Evaluation

5.1 Simulation settings

As a simulation environment, we assume that LoRaWAN where five gateways are placed at random locations, and if data from a node is successfully delivered to any one of the gateways, the reception is successful. Each LoRa node generates data periodically and sends it to the gateway immediately after it is generated along with its location information. The data transmission is done by ALOHA protocol and no re-transmission is done.

Based on the node location information, the gateway obtains the number of nodes belonging to each sub-area, and determines the spreading rate to be assigned to the nodes belonging to each sub-area by the GA. After determining the allocation method, a control signal is sent to each sub-area by broadcast, and the node that receives the message changes its own spreading factor. The area is divided into 10×10 sub-areas, and nodes in the same area are assigned the same spreading factor. To demonstrate the adaptability of the proposed method to various situations, we design a scenario with the following three policies as the environmental setting for the simulation. The details of the scenarios are described in the subsequent sections, corresponding to each of them.

- Evaluation in a situation where the optimal solution changes due to node mobility
- Evaluation in a situation where the optimal solution changes due to gateway failure

As the fitness of data arrival, we use the average value of data arrival rate calculated by Eq. (9), and use Eq. (11) as the fitness for power consumption. The evaluation function is set to be the sum of these two fitness functions. As methods for comparison, we use a method with FTDGA replaced by a simple genetic algorithm (SGA) or a method replaced by TDGA that does not use the feedback function $\mathcal{T}(t)$. In addition to the algorithm described in Section 3.1, the compared SGA used here adopts an elite strategy in which individuals with good evaluation values are kept as elites in the population, and the number of elites to be kept is 40.

The parameters for GA are shown in Table 2, and other simulation parameters are shown in Table 1.

Table 1: Simulation settings

Parameter	Value
Simulation time	10000 s
Period of calling GA	50 s
First GA call time	100 s
Power limit	118.8 mW

Table 2: Parameters for GA

Parameter	Value
Size of population	500
Number of generations to calculate for each GA call	100
Mutation rate	0.05
Crossover rate	0.3

The parameters related to LoRa communication are shown in Table 3.

Table 3: Parameters for LoRa

Parameter	Value	details
FIELD_WIDTH	10 km	Width of the area to be controlled
FIELD_HEIGHT	10 km	Height of the area to be controlled
AREA_DIV_X	10	Number of horizontal separations of the area
AREA_DIV_Y	10	Number of vertical separations of the area
DATA_RATE	100 s	Data generation period

The probability of errors in the signal reaching the gateway changes depending on the distance between the node and the gateway and the spreading factor. In this paper, we use the values of packet error rate shown in Table 4 for simulation.

5.2 Simulation scenarios

5.2.1 Node mobility

In this scenario, the simulation assumes that a part of nodes moves. Due to the nature of spreading factor, using low spreading factors near the gateway and using high spreading factors far from the gateway improves the data arrival rate, so significant changes of the optimal solution almost never cause with the proposed method when the nodes move. Specific environmental changes are as follows.

- In an area of 10 km square, 600 nodes will be placed at random locations. These

Table 4: Available spreading factors and packet error rate

distance\SF	7	8	9	10	11	12
0–1km	0%	0%	0%	0%	0%	0%
1–2km	20%	10%	0%	0%	0%	0%
2–3km	40%	30%	15%	15%	10%	10%
3–4km	50%	40%	30%	25%	20%	15%
4–5km	60%	55%	45%	30%	25%	20%
5km–	80%	70%	60%	50%	40%	35%

nodes do not move.

- After that, 20 mobile nodes will be added to each of 20 randomly chosen sub-areas.
- Every 2500 seconds, these added nodes move to another nearby sub-area. At this time, the destination of the 20 nodes in the same sub-area is assumed to be the same.

5.2.2 Gateway failure

In this scenario, the simulation assumes that gateways become failure. When a gateway fails, the optimal solution changes further than the scenario in the previous section because the nodes that were using lower spreading factors near the gateway are forced to use higher spreading factors to communicate with other gateways. Specifically, every 2000 seconds during the simulation, one gateway is selected randomly and failed. Communication between nodes and the failed gateway becomes impossible.

5.2.3 Balancing power consumption of individual nodes

We use another fitness function for the goal of equalizing the power consumption of individual nodes. The value of fitness for this goal changes over time.

The fitness function for balancing the power consumption of nodes is defined as $F_{pow-each}$ in Eq. (12).

$$F_{pow-each} = -\frac{pow_{sum} - pow_{min}}{pow_{max} - pow_{min}} \tag{12}$$

In Eq. (12), the node with the minimum remaining battery power in each sub-area when the time of one transmission cycle has elapsed when using the spreading factor assignment in the individual to be calculated is obtained, and the sum of the difference between the minimum remaining battery power in those nodes and the remaining battery power in the other nodes is set to pow_{sum}. In Eq. (12), pow_{max} and pow_{min} are the values when individuals are given such that pow_{sum} is the maximum and minimum, respectively.

5.3 Results

We performed ten simulations for each single parameter setting while changing the target entropy H^* and feedback gain τ of FTDGA. In all simulations, the environment the same, i.e., the placement of nodes, destinations of moving nodes, and the placement of gateways, are the same. For reference, SGA with the elite number of 40 and TDGA were also run with the same environment settings. GA runs every 50 seconds, and the individual with the highest fitness value will be output as the solution. The parameters used are as shown in Table 5.

Table 5: Parameters for TDGA and FTDGA

Parameter	Value
T	$10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$
H^*	$20, 40, 80$
τ	$0.01, 0.1, 1$

5.3.1 Nodes mobility

The distribution of fitness obtained as a result of SGA and TDGA is represented by box plots in Fig 3, and the results of FTDGA are shown in Fig. 4.

First of all, with TDGA, we can see that for $T = 10^{-4}$ and $T = 10^{-3}$, the solution does not exceed the upper power limit in more than half of the simulations, but for lower values of T, the solution may exceed the upper power limit and produce a solution with

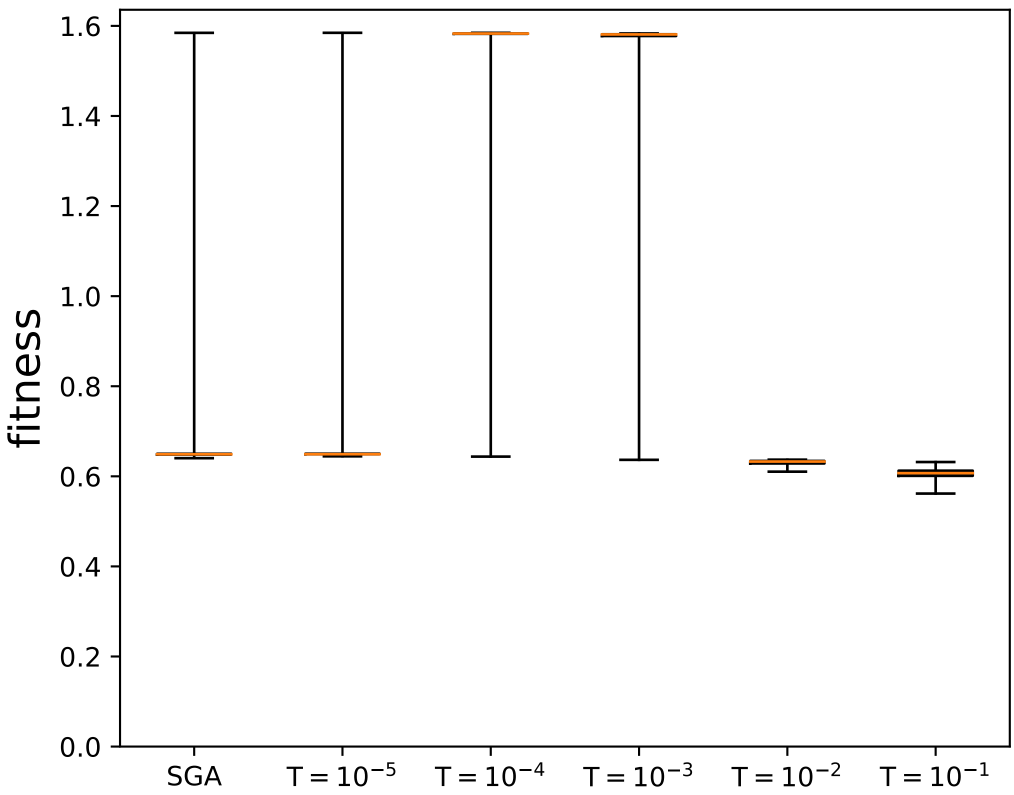

Figure 3: Fitness comparison with TDGA (Node mobility)

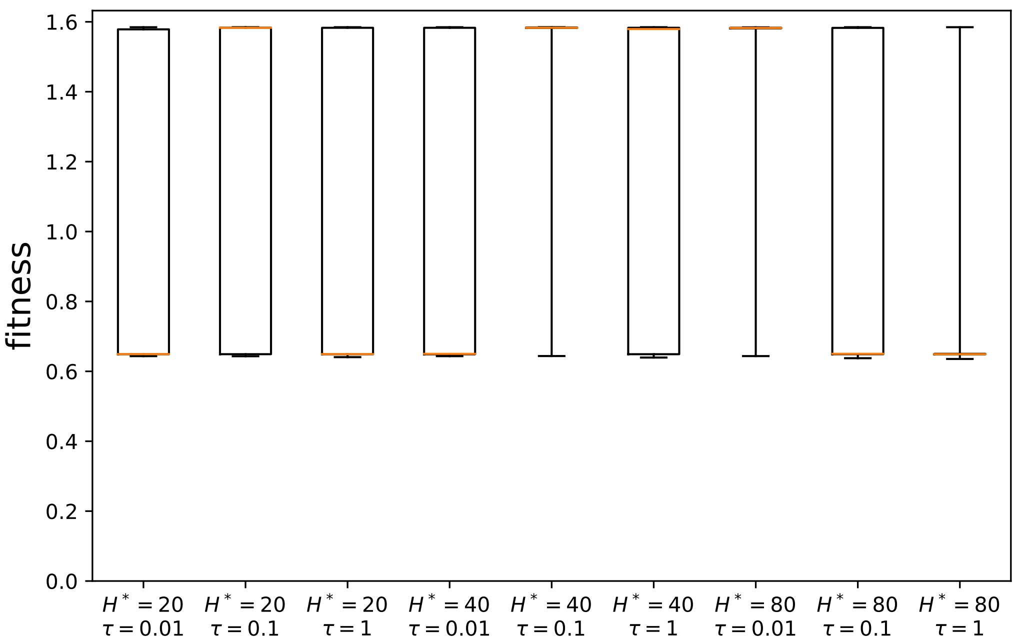

Figure 4: Fitness comparison with FTDGA (Node mobility)

low fitness. However, for values of T less than that, the power limit is exceeded and a solution with low fitness is output. For T values higher than these, we were not able to find any solution that does not exceed the upper power limit. In other words, neither too low nor too high temperature parameters can quickly find solutions with high fitness.

With FTDGA, the entropy immediately after the start of the simulation was lower than TDGA because FTDGA tried to maintain the target entropy H^* for $H^* = 20$, $\tau = 0.01$ and $H^* = 40$, $\tau = 0.01$, the initial solution finding performance was not sufficient. For $H^* = 40$, $\tau = 0.1$ and $H^* = 80$, $\tau = 0.01$, the performance is close to that of TDGA. However, when τ is further increased beyond these values, the change in entropy H becomes too large, and as shown in Fig. 6, the value of H fluctuates around zero or close to the upper limit, resulting in a decrease in performance. For smaller values of τ, the feedback function has almost no effect on the temperature and the behavior shown in Fig. 8 is almost the same as that of TDGA shown in Fig. 7.

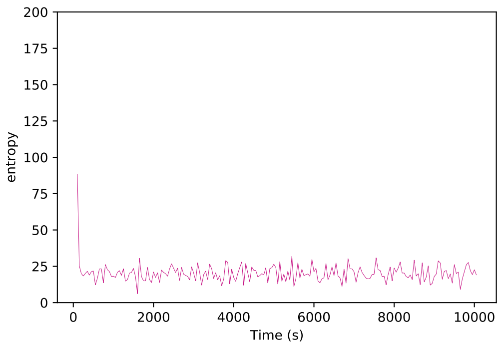

Figure 5: Entropy with $H^* = 20$ and $\tau = 0.01$

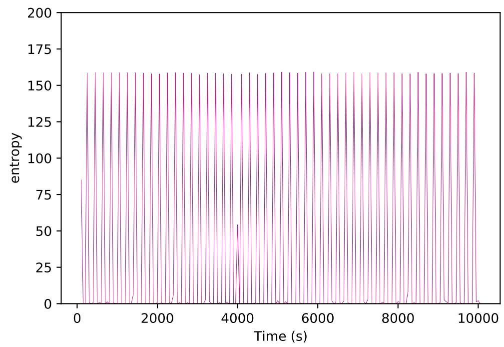

Figure 6: Entropy with $H^* = 40$ and $\tau = 1$

5.3.2 Gateway failure

The resulting fitness distributions for SGA and TDGA are represented by box plots in Fig 9, and the results for FTDGA are shown in Fig 10.

With TDGA, as in the environment of the previous section, the fitness of the most output solution was superior when $T = 10^{-3}$. With FTDGA, the fitness of the most output solution was also superior when $H^* = 40$, $\tau = 0.1$ settings, as in the previous

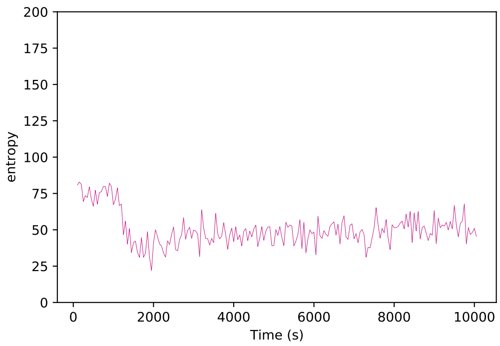

Figure 7: Entropy with $T = 10^{-4}$

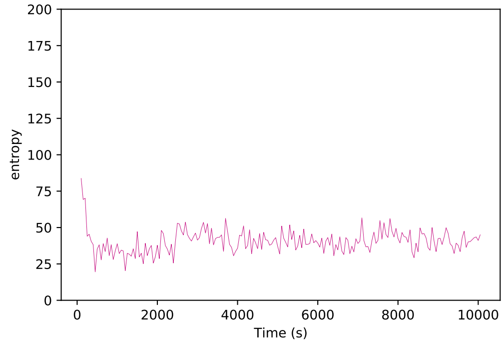

Figure 8: Entropy with $H^* = 40$ and $\tau = 0.001$

section.

5.3.3 Balancing power consumption of individual nodes

In this section, the half of the fitness function for power consumption is set to $F_{pow-each}$ and the other half of it is $F_{pow-total}$. The distribution of fitness obtained as a result of SGA and TDGA is represented by box plots in Fig. 11, and the results of FTDGA are shown in Fig. 12.

With TDGA, the appropriate temperature T was 10^{-3} or 10^{-4} in the previous environment. But with this environment setting, the performance was better at $T = 10^{-4}$, while the solution satisfying the power constraint was not found at $T10^{-3}$. With this setting, the performance was better at $T = 10^{-4}$, while $T = 10^{-3}$ did not find a solution that satisfied the power constraint, indicating that the performance was better at $T = 10^{-5}$ where the temperature was lower.

On the other hand, FTDGA performs almost as well as the other settings at the setting of $H^* = 40$ and $\tau = 0.1$, which was the best in the two previous environments.

5.4 Discussion

In our proposed method using TDGA, when selecting individuals, it sequentially selects one by one the individuals to be left for the next generation. When selecting the first few individuals, the higher the temperature T, the more heterogeneous individuals will

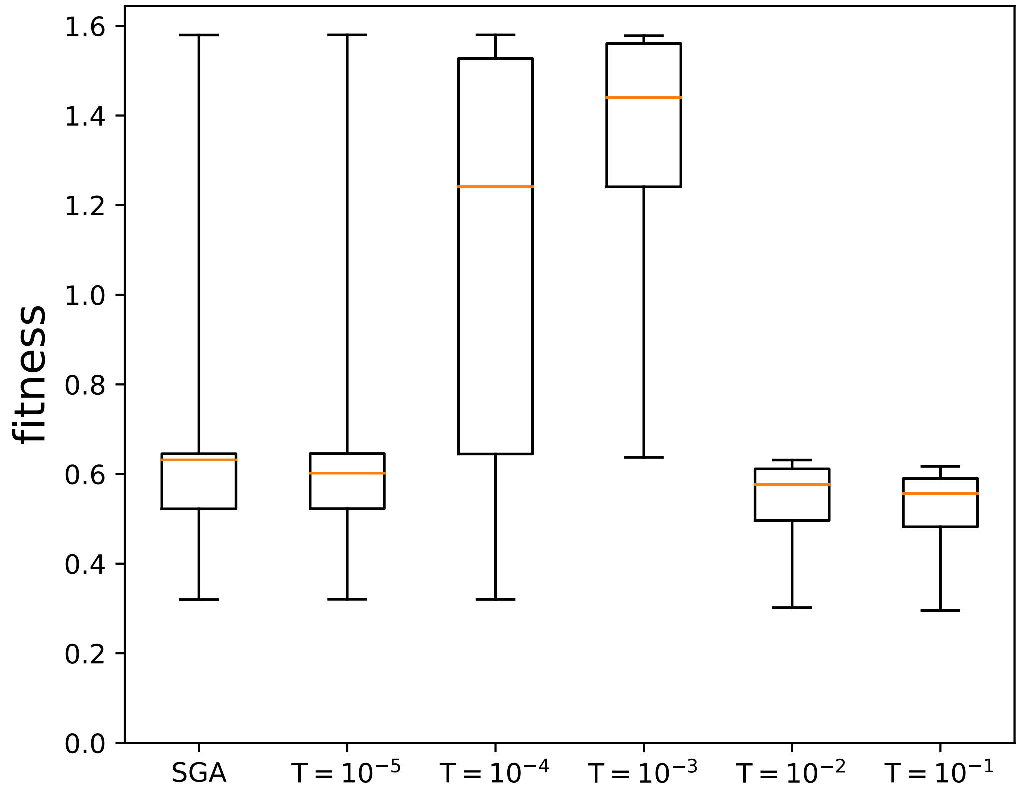

Figure 9: Fitness comparison with TDGA (Gateway failure)

be selected as the next generation population, because there is a tendency to try to leave individuals that are significantly different from the previously selected individuals in order to increase the entropy value to some extent in order to minimize the free energy F. Because of this nature of the selection operation of TDGA, it is likely that a certain value of entropy in FTDGA will be sufficient to achieve sufficiently good fitness in most environments to be attainable in the design of the gene expression.

Moreover, since FTDGA can automatically set T, we believe that it can be used as a method to search for T when the appropriate temperature is not known. In this case, it is necessary to know the appropriate target entropy H^* , but this is not very dependent on the environment, so it is easier to set the parameters H^* and τ than trying different temperatures with TDGA.

Regarding the relationship between fitness and actual performance, first of all, for power, the higher the $F_{pow-total}$ and $F_{pow-each}$ definitions, the more power can be saved.

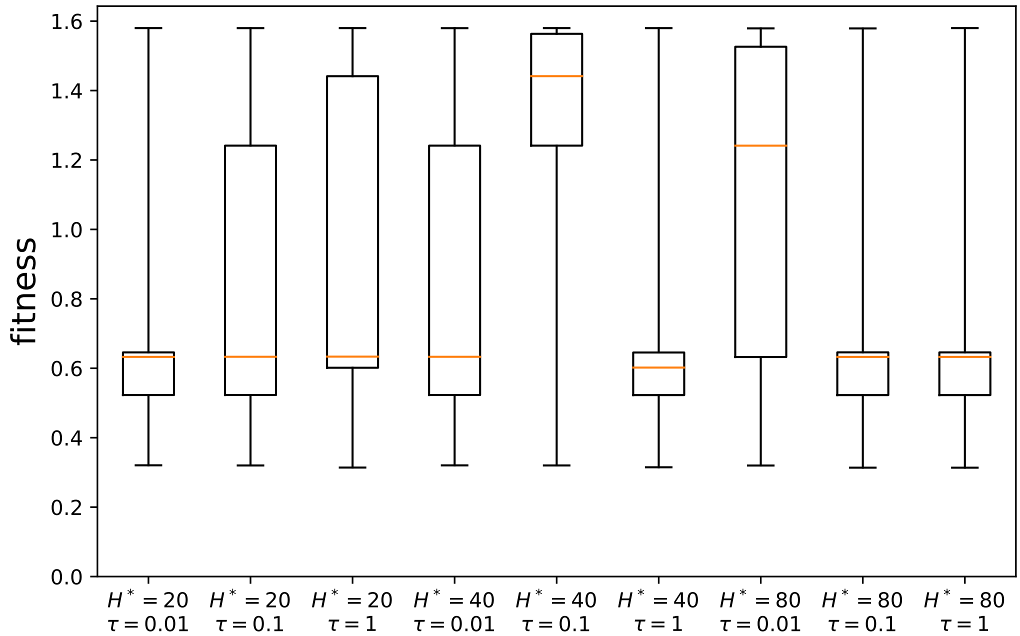

Figure 10: Fitness comparison with FTDGA (Gateway failure)

As for throughput, the actual data collection rate (shown by the green line) during the simulation and F_{thr} (shown by the green line) are almost identical as shown in Fig. 13. The data extraction rate (DER) is the ratio of packets sent by the terminal to those received by the gateway during a certain period of time. The calculation range of DER in Figure 13 is 500 seconds.

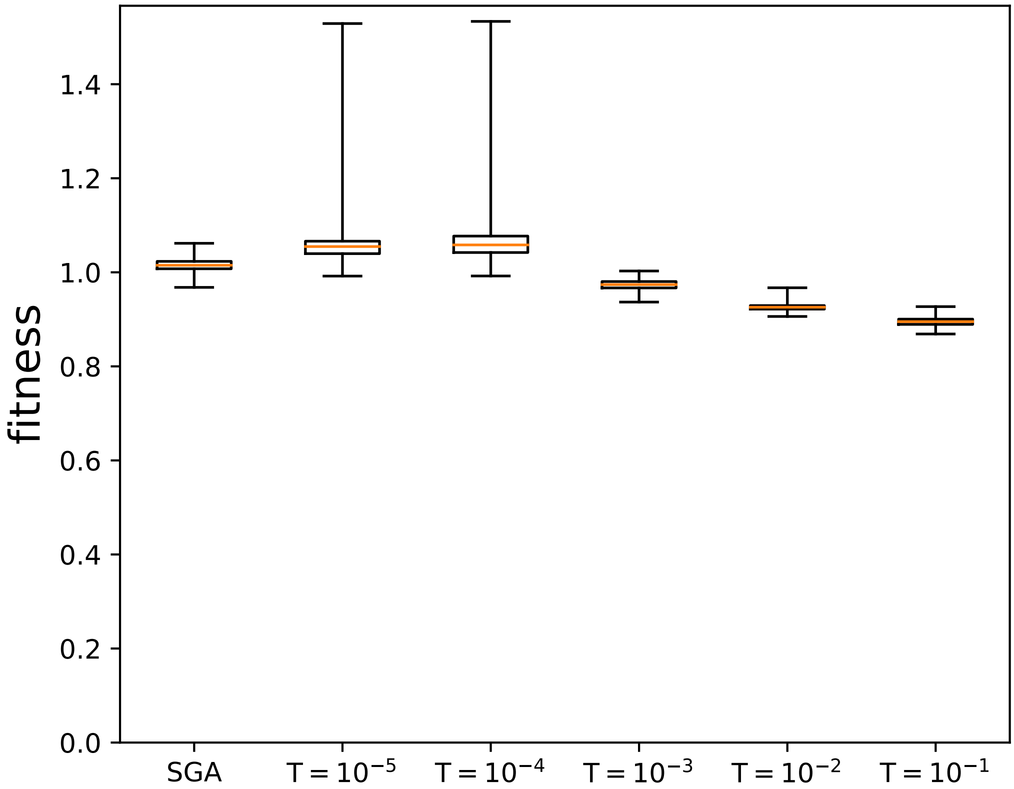

Figure 11: Fitness comparison with TDGA (Balancing power consumption)

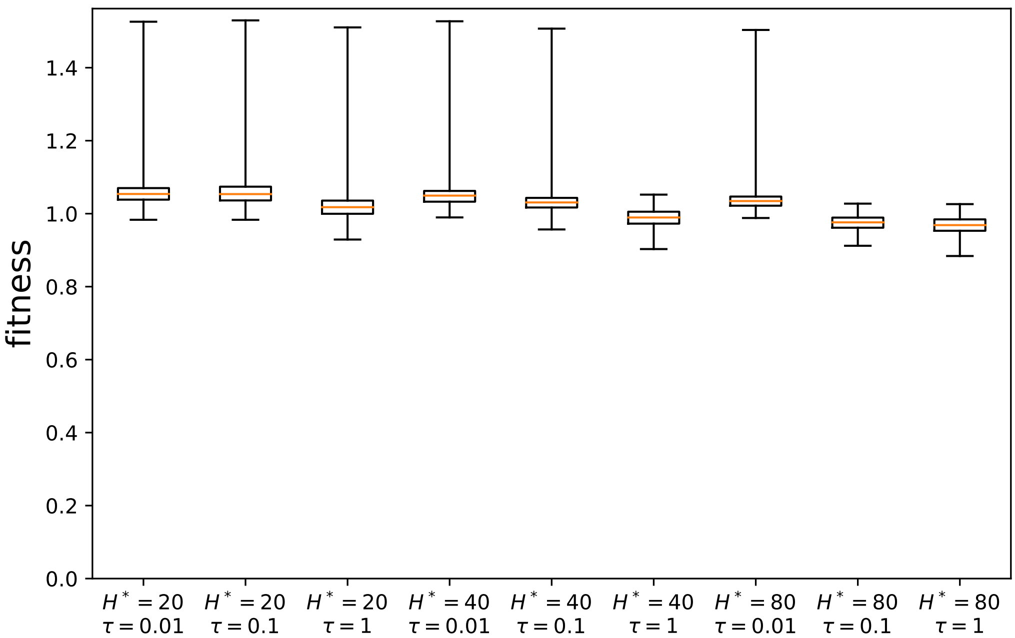

Figure 12: Fitness comparison with FTDGA (Balancing power consumption)

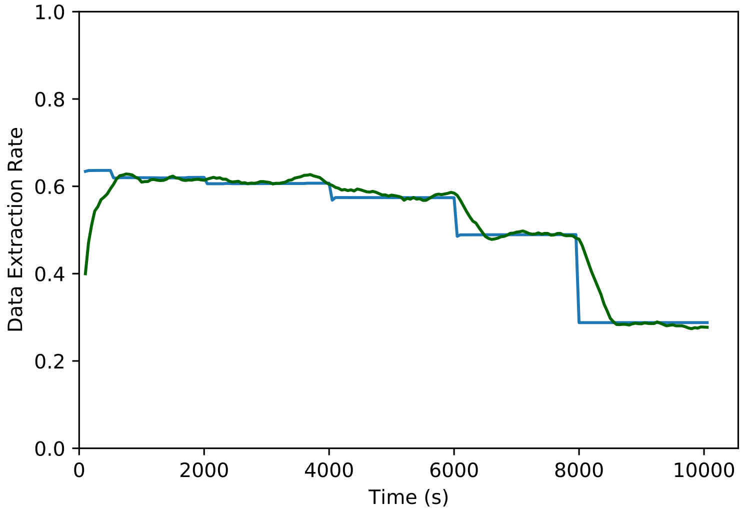

Figure 13: Data extraction rate and F_{thr} ($H^* = 40$ and $\tau = 0.1$)

6 Conclusion

LoRaWAN is a kind of LPWA communication technology targeting IoT networks, which has a higher degree of freedom in development than other LPWA technologies, and has become a major research target for IoT networks. LoRaWAN can dynamically change the data rate and transmission power of devices using control packets, and control them according to the network conditions. However, since the data rate of LoRaWAN is low, ranging from several hundred bps to several tens of kbps, it is more difficult to obtain sufficient information on the network required than conventional wireless networks, making it difficult to perform the optimal control according to environmental changes.

In this thesis, we proposed and demonstrated the effectiveness of a method for appropriately controlling the spreading factor to simultaneously improve the throughput and energy consumption of LoRaWAN nodes in a fluctuating environment. By using FTDGA, which uses the evolutionary mechanism of living organisms and keeps the diversity of populations, to determine the spreading factor assigned to each node, we show that it is possible to control the spreading factor to rapidly follow environmental changes such as changes in the distribution of the number of nodes and the failure of gateway nodes. We also show that keeping the diversity of the population in our FTDGA based scaling factor allocation method has an advantage to design parameters. Given an appropriate diversity in FTGDA, it is possible to obtain higher fitness than SGA, and the same level of fitness as TDGA with sufficient parameter tuning.

Future work includes the development of a qualitative method to determine the appropriate target entropy H^* and feedback gain τ. We expect that these parameters are determined once the gene encoding is determined.

Acknowledgements

I would like to express my deepest gratitude to Professor Masayuki Murata of the Graduate School of Information Science and Technology, Osaka University, for his encouragement and guidance during the course of this research. I would also like to express my sincere gratitude to Assistant Professor Daichi Kominami of the Graduate School of Information Science and Technology, Osaka University, for his many hours of advice and guidance on the direction of my research. Moreover, I also would like to thank Hideyuki Shimonishi of NEC. He gave me insightful comments and suggestions. I am grateful to Associate Professor Shinichi Arakawa of the Graduate School of Information Science and Technology, Osaka University, and Associate Professor Yuichi Ohsita of the Institute for Open and Transdisciplinary Research Initiatives, Osaka University, for their extensive advice. Finally, I would like to express my gratitude to Mr. Masaaki Yamauchi, Ms. Shiori Takagi, and other members of Murata Laboratory for their continuous support.

References

[1] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14 – 21, Mar. 2017.

[2] LoRa Alliance, "LoRaWAN what is it? A technical overview of LoRa and LoRaWAN," accessed:2021/12/17. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf

[3] M. Saari, A. M. bin Baharudin, P. Sillberg, S. Hyrynsalmi, and W. Yan, "LoRa – A survey of recent research trends," in *Proceedings of 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2018, pp. 872 – 877.

[4] LoRa Alliance, "LoRaWAN specification (v1.1)," accessed:2021/12/17. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf

[5] M. Bor, J. Vidler, and U. Roedig, "LoRa for the internet of things," in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN'16. Junction Publishing, Feb. 2016, pp. 361 – 366.

[6] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology," in *Proceedings of 2015 14th International Conference on ITS Telecommunications (ITST)*, Dec. 2015, pp. 55 – 59.

[7] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Nov. 2016, pp. 59 – 67.

[8] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in LoRaWAN: Recent solutions and major challenges," *Sensors*, vol. 20, no. 18, p. 5044, Sep. 2020.

[9] Y. Fujita, D. Kominami, H. Shimonishi, and M. Murata, "An adaptive spreading factor allocation method to dynamic environmental changes using thermodynamical genetic algorithm for LoRaWAN," *IEICE Tech. Rep*, vol. 121, no. 263, pp. 29 – 34, Nov. 2021.

[10] N. Mori, H. Kita, and Y. Nishikawa, "Adaptation to a changing environment by means of the thermodynamical genetic algorithm," *Transactions of the Institute of Systems, Control and Information Engineers*, vol. 12, no. 4, pp. 240 – 249, Apr. 1999.

[11] G. Park, W. Lee, and I. Joe, "Network resource optimization with reinforcement learning for low power wide area networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, p. 176, Sep. 2020.

[12] S. Narieda, T. Fujii, and K. Umebayashi, "Energy constrained optimization for spreading factor allocation in LoRaWAN," *Sensors*, vol. 20, no. 16, p. 4417, Aug. 2020.

[13] A. Loubany, S. Lahoud, and R. El Chall, "Adaptive algorithm for spreading factor selection in LoRaWAN networks with multiple gateways," *Computer Networks*, vol. 182, p. 107491, Aug. 2020.

[14] M. Hamnache, R. Kacimi, and A.-L. Beylot, "L3SFA: Load shifting strategy for spreading factor allocation in LoRaWAN systems," in *2020 IEEE 45th Conference on Local Computer Networks (LCN).* ieee, Nov. 2020, pp. 216 – 224.

[15] J.-T. Lim and Y. Han, "Spreading factor allocation for massive connectivity in LoRa systems," *IEEE Communications Letters*, vol. 22, no. 4, pp. 800 – 803, Jan. 2018.

[16] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, "Towards energy-fairness in LoRa networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Jul. 2019, pp. 788 – 798.

[17] M. Asad Ullah, J. Iqbal, A. Hoeller, R. D. Souza, and H. Alves, "K-means spreading factor allocation for large-scale LoRa networks," *Sensors*, vol. 19, no. 21, p. 4723, Oct. 2019.

[18] N. Mori, J. Yoshida, H. Kita, and Y. Nishikawa, "A thermodynamical selection rule in the genetic algorithm," *Transactions of the Institute of Systems, Control and Information Engineers*, vol. 9, no. 2, pp. 82 – 90, Feb. 1996.

[19] N. Mori, Y. Yabumoto, H. Kita, and Y. Nishikawa, "Multi-objective optimization by means of the thermodynamical genetic algorithm," *Transactions of the Institute of Systems, Control and Information Engineers*, vol. 11, no. 3, pp. 103 – 111, Mar. 1998.

[20] Semtech, "Semtech SX1276," https://www.semtech.com/products/wireless-rf/lora-core/sx1276, accessed: 2022-1-17.