Distributed Light Brightness Control based on cuSASGP

Keiko Ono
Department of Electronics and Informatics,
Ryukoku University, Japan
Email: kono@rins.ryukoku.ac.jp

Yoshiko Hanada
Faculty of Engineering Science,
Kansai University, Japan
Email: hanada@kansai-u.ac.jp

***Abstract*—This paper describes an approach for controlling light illuminance using the CUDA-based Self-adaptive Subpopulation Model of Genetic Programming (cuSASGP). The method involves the evolution of a genetic programming lighting control rule for the ceiling lights in an office room to satisfy different brightness requirements at each desk and reduce electric power consumption. Although the lighting control problem has many local minima, cuSASGP uses solution features to construct an appropriate island formation that avoids these local minima. Thus, an approach for controlling light illuminance based on cuSASGP could be expected to improve performance in terms of avoiding local minima and genetic diversity. We first define the lighting control problem for ceiling lights, and propose a genetic programming approach. We implement five types of functional nodes and three types of terminal nodes. Moreover, we verify that genetic diversity can be achieved by adopting subpopulation models such as the island method and cuSASGP in the lighting control problem. For two different environments, we demonstrate that the proposed genetic programming approach can optimize an appropriate lighting pattern that satisfies both user requests and energy constraints, and that use of cuSASGP enhance genetic diversity.**

I. Introduction

In recent years, it has been reported that individual lighting control is related to intellectual productivity[1], [2] and that intellectual productivity often increases when a user's brightness requirements are satisfied. Moreover, individual control is expected to reduce the energy consumed by lighting appliances[3], which accounts for more than 20% of energy consumption in office buildings. To control lights individually, an algorithm must assign a different output value to each light[4], [5], [6], [7], [8]. In general, there exist various lighting output values that satisfy each user's desired brightness; however, these values rarely minimize the energy consumption. For example, when a user require their desk to be lit very brightly, the user's overhead light could output half of its maximum brightness, or a corner light located away from the user could output its maximum power. Therefore, we can consider the lighting control to have a number of local minima.

Genetic Programming (GP) is a widely used practical optimization method. It is often used to solve problems involving local minima, and has been used to produce solutions that are competitive with human-generated results[9]. Numerous researchers have applied GP to a variety of fields, including electrical circuits, robotics and image filters[10], [11]. In this paper, we define the problem of individual lighting control based on GP, and present an effective method to escape from the inherent local minima.

GP commonly employs a parallel model to avoid premature convergence. A parallel model GP frequently gives better solutions than single population models. The island model is a representative parallel GP method. The island model forms subpopulations each of which executes GP searches in parallel to minimize an objective function. The subpopulations have the same objective function, and some individuals often migrate from one subpopulation to another. Migration strategies can apply ring, grid, and random topologies, and the elite individuals that have better fitness than others in a subpopulation are often selected to migrate. As mentioned above, the island model encourages genetic diversity by generating subpopulations, but we hypothesize that the genetic diversity will decrease if elite individuals are exchanged repeatedly. Therefore, we have proposed an adaptive migration topology that considers the features of each individual, namely the CUDA-based Self-adaptive Subpopulation Model of Genetic Programming (cuSASGP)[12]. In this previous study, well-known benchmark problems were evaluated to confirm that the adaptive migration topology based on feature such as the fitness and size of an individual enhances genetic diversity and improves performance.

In this paper, we consider the problem of lighting control with multiple local minima, and propose a GP method that automatically generates lighting control rules. Moreover, to enhance the genetic diversity, we apply cuSASGP to the proposed lighting control method, and verify the effectiveness of this approach.

The reminder of this paper is organized as follows. In Section II, we formulate the optimization problem discussed in this paper and explain our parallel GP method. In Section IV, we briefly describe the CUDA-based adaptive subpopulation model in GP. Section V-C examines the effectiveness of the proposed method through experiments using two different brightness patterns. We first explain how the proposed method is implemented, and then show that the GP approach for the distributed light control problem can control each light's output to satisfy the desired lighting pattern. Moreover, we introduce six comparative methods in order to analyze the effectiveness of the proposed method, and verify that incor-

978-1-5090-0623-6/16/$31.00 ©2016 IEEE

porating cuSASGP enhances genetic diversity over the island model GP.

II. Problem Definition and Parallel Genetic Programming

We consider the problem of controlling the brightness of individual ceiling lights to satisfy multiple illuminance requirements and reduce energy consumption. We refer to this problem as the Distributed Light Control (DLC) problem in this paper. More specifically, under an environment consisting of I ceiling lights equipped with controllable bulbs and M illuminance sensors, we consider the problem of minimizing a function $f(\boldsymbol{L})$ in terms of the output values $\boldsymbol{L}$ for each light. This function consists of two kinds of items, i.e.,

$$f(\boldsymbol{L}) = \alpha \cdot \phi_1 \sum_{m=0}^{M} |G_m^T - G_m^O| + \phi_2 \sum_{i=0}^{I} W_i, \tag{1}$$

where G_m^T $(m = 1, \ldots, M)$ is the target illuminance value, G_m^O is the current illuminance value, W_i $(i = 1, \ldots, I)$ is the energy consumption of the ith light,

and w indicates a weight value. Let L_i denote the output of the ith light, i.e., $\boldsymbol{L} = \{L_o, L_1, \ldots, L_I\}$. The illuminance is the total luminous flux incident on a surface per unit area. The, the illuminance of the mth sensor S_m $(m = 1, \ldots, M)$ is obtained by

$$S_m = \sum_{i=0}^{I} g_{m,i}(L_i), \tag{2}$$

where the function $g_{m,i}$ returns the illuminance at sensor S_m when the output value of L_i is L_i^O. The energy consumption of L_i is obtained by

$$W_i = w_i(L_i). \tag{3}$$

In equation (1), each item deals with different kinds of physical values, so we use ϕ_1 and ϕ_2 as normalization parameters, and use α as a weighting parameter.

Next, we recall some aspects of parallel GP. In parallel GP, the total population $P = x_j;\ \ j = 1, \ldots, J$ is partitioned into U subpopulations by

$$P = \bigcup_{u=1}^{U} V^u,$$

and the subpopulations execute parallel GP searches with the aim of minimizing $f(\boldsymbol{L})$, where x_i indicates an individual. In GP, each individual x_j is expressed in terms of a tree, and so the output values $\boldsymbol{L}$ can be determined by tracing x_j. In each fixed cycle, some individuals are exchanged through the process of migration. Of the several migration topologies available, the ring topology is the most popular.

III. Genetic Programming Approach for lighting control

In the DLC system, each user sets a different target illuminance value, and an optimizer determines the optimal output values $\{L_o^*, L_1^*, \ldots, L_I^*\}$. However, if there are various optimum brightness patterns that give the required illuminance and lower energy consumption, mbox$\boldsymbol{L^*}$= $\{L_o^*, L_1^*, \ldots, L_I^*\}$ may not be the global optimum[13], [14], [15]. In general, meta-heuristic methods such as Genetic Algorithms and Simulated Annealing can be applied to escape from local minima. GP is another useful meta-heuristic algorithm for exploring optimal solutions using evolving computer programs or rules. In the DLC problem, we hypothesize the existence of important rules for effectively controlling the brightness of lights, and assume that GP is suitable for determining a solution. Therefore, in this paper, we apply GP to the DLC system. First, we model the GP approach to the DLC problem. In Section IV, we attempt to deliver further performance improvements by proposing a Light Brightness Control method based on cuSASGP.

To apply GP to real-world problems, we must define the function nodes and terminal nodes that compose the solution. In general, a function node executes a conditional branch and a sequential process, whereas terminal nodes execute the real actions of the target system. In the DLC problem, we assume there is a need for conditional branches based on each sensor value and actions that change the output value of each light. More specifically, we use the difference between the target illuminance value S_m^T and the current illuminance value S_m^O as the judgement criterion for conditional branches, and apply three kinds of actions to terminal nodes: retain the current output value L_i, raise the current output value L_i, or lower the current output value L_i. We define function nodes and terminal nodes in Table I and II . Table I presents an abstraction of the function nodes, where $D_{S_m} = |S_m^T - S_m^O|$ and a_1, a_2 are functional arguments. For example, node Illum$_{S_m}^{10}$ is a conditional branch that executes a_1 or a_2 according to whetherD_{S_m} is less than 10[lx]. When D_{S_m} is less than 10[lx], a_1 is executed and vice versa. We assign three values to D_{S_m} (10 [lx], 20 [lx], and 30 [lx]) for the following reasons. Illuminace values of less than 10[lx] cannot be determined particularly well by the human visual system, so we can consider the target illuminance to be required when D_{S_m} is less than 10 [lx]. Moreover, we feel that the current illuminance is slightly lower or higher than the target illuminance when D_{S_m} is in the range 10–20 [lx], but these values are often acceptable. When D_{S_m} is more than 30 [lx], we notice the difference between the current illuminance and the target illuminance. Therefore, we define three values of D_{S_m}. In the DLC problem, there are M sensors. Thus, we have M kinds of functional nodes with Illum$_{S_m}^{10}$, Illum$_{S_m}^{20}$ and llum$_{S_m}^{30}$. When $M = 4$, the number of functional nodes is $4 \times 3 + 2 = 14$.

TABLE I
Functional Nodes

Functional Node	Function
Illum$_{S_m}^{10}$ (a_1, a_2)	If D_{S_m} is less than 10, execute a_1, otherwise execute a_2.
Illum$_{S_m}^{20}$ (a_1, a_2)	If D_{S_m} is less than 20, execute a_1, otherwise execute a_2.
Illum$_{S_m}^{30}$ (a_1, a_2)	If D_{S_m} is less than 30, execute a_1, otherwise execute a_2.
Prog2(a_1, a_2)	Execute a_1 and a_2 in order.
Prog3(a_1, a_2, a_3)	Execute a_1, a_2 and a_3 in order.

We define the three kinds of terminal nodes described in Table

TABLE II
TERMINAL NODES

Terminal Node	Function
Up_{L_i}	Raise output value β of Light L_i
Down_{L_i}	Lower output value β of Light L_i
Keep_{L_i}	Do not change output value of Light L_i

II, where the range for raising of lowering output is controlled by the parameter β. We describe how to set the length in Section V-B. As for the functional nodes, there are M kinds of terminal nodes for each Up_{L_i}, Down_{L_i} and Keep_{L_i}, so when $M = 4$, the number of terminal nodes is $4 \times 3 = 12$. Figure 1 illustrates an example of a solution. The solution has a depth of 2, and there exist three kinds of functional node and four kinds of terminal node. The arguments a_1, a_2 of $\mathrm{Illum}^{10}_{S_1}$ are $\mathrm{Prog2}(a_1, a_2)$ and $\mathrm{Illum}^{20}_{S_3}$, and those of $\mathrm{Prog2}(a_1, a_2)$ are Keep_{L_2} and Up_{L_1}, respectively.

Next, we explain how to evaluate the given solution. We evaluate the solution according to the Santa Fe ant trail problem, which is sometimes used as a representative problem in GP. The object of the Santa Fe ant trail problem is to automatically find the optimal program to control artificial ants that efficiently collect food within a certain region. The region is a 32 ×32 grid in which there are 89 items of food. Ants start from a root node in the upper-left cell with initial energy E, and trace a solution according to a depth-first search. While searching for a terminal node, the artificial ants move or change their direction, and gradually lose energy. If there is food at a particular location, the ants consume the food and gain energy. Once an ant's energy E has vanished, the search is terminated. In the Santa Fe ant trail problem, a solution indicates a rule controlling the artificial ants. In the same way, the object of the DLC problem is to optimize a rule for lighting control. Therefore, the same framework can be used to find solutions to the DLC problem. More specifically, the initial output of light $\boldsymbol{L}$ is set to its maximum (8-bit) value of 256. Analogously to the ants' energy, the number of iterations for tracing a solution is set to E. While E remains positive, we trace a solution according to a depth-first search. When we reach a terminal node, E is set to $E - 1$. After tracing a solution, we have the output value for $\boldsymbol{L}$. Based on the current $\boldsymbol{L}$, the objective function in equation (1) is evaluated, where the illuminance in equation (2) is estimated by an illuminance simulator constructed in a real experimental environment[1] .

IV. cuSASGP

The most common way to enhance genetic diversity is to apply parallel GP models. In the island model, which is a representative parallel GP model, elite individuals with higher fitness values periodically migrate to the next island, while the topology of the islands remains. We hypothesized that the repeated migration of elite individuals that are selected solely in terms of fitness would result in each island containing a similar elite population. Therefore, to enhance genetic diversity, we proposed the cuSASGP approach[12], which generates a similarity network in terms of features such as fitness and tree size, and adaptively changes the topology of subpopulations to maintain genetic diversity. We now briefly explain the cuSASGP in terms of migration, where we use CUDA in the steps C3 and C4.

[1]Details of the lluminance simulator are given in Section V-B.

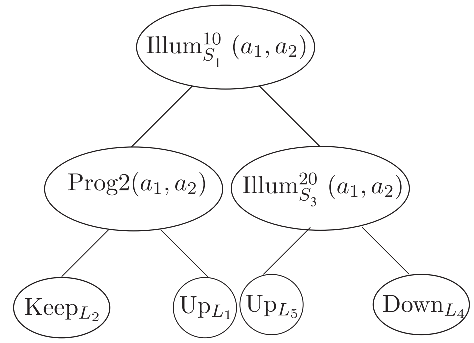


Fig. 1. Example of an individual solution.

- C0: If g is a reconstruction generation, then perform Steps C1–C7.
- C1: Send V^u to subpopulation #1 .
- C2: If the subpopulation is #1, then perform Steps C3–C6; otherwise, perform Step C7.
- C3: Evaluate the similarity between x_i and $x_j (i \neq j)$ in terms of features.
- C4: Generate a weighted network based on these features.
- C5: Divide the population P into U subpopulations by network clustering, and create $\hat{V}^u$.[2].
- C6: Send $\hat{V}^u$ to subpopulation #u.
- C7: Replace V^u with $\hat{V}^u$.

In [12], we constructed a weighted network based on fitness and node size. We know there are many features that can express the relationship between individuals, but, in this paper, we focus on the DLC problem based on cuSASGP to enhance genetic diversity. Therefore, we use the fitness and node size as a first step. Thus far, we have verified the performance of cuSASGP using the well-known GP benchmark problems of symbolic regression problems. As the DLC problem has multiple local minima, maintaining diversity tends to improve performance. Moreover, when we deal with the DLC problem using GP, the cuSASGP framework is known to be effective. To enhance genetic diversity, we apply cuSASGP to the DLC problem, and demonstrate the difference between island GP and cuSASGP in terms of the resultant genetic diversity.

[2]There are a lot of clustering methods to create subpopulations. In [12], we used the Newman Clustering that is the common method to create communities based on a weighted network. We referred to communities as subpopulations.

V. Experimental Evaluation

Through experiments using static and dynamic illuminance environments, we evaluated the effectiveness of the proposed method. In the dynamic environment, the illuminance value required by each user changes during the search.

A. Comparison Methods

The proposed method evaluates equation (1) to optimize the output value of light $\boldsymbol{L}$ so as to satisfy the required illuminance values B^T and reduce energy comsumption. To analyze the influence of the weight parameter α, we introduce five methods that change the optimal balance between the required illuminance values and the energy consumption. We set α to 1.0, 3.0, 5.0, 7.0, and 9.0 in turn, and refer to these methods as Proposed-α.

A simple GP approach for the DLC problem is to execute one-point crossover and optimize each individual sequentially. We introduce this method as a baseline for evaluating the effectiveness of GP for the DLC problem. We refer to this as the Simple method. Moreover, a common way to enhance genetic diversity in GP is to apply a subpopulation model. We used an island model with a ring topology an one-point crossover as a subpopulation model. We refer to this method as the Island method. The topology of the Island model does not change during the search, so the number of individuals in each subpopulation is fixed.

B. Experimental Setting

To evaluate the effectiveness of the proposed method, we built the experimental room shown in Fig.2. This room contains eight ceiling lights and three illuminance sensors set on a desk. Using this room, we constructed an illuminance simulator to determine the current illuminance of equation (2). The 8-bit lights can can change their output value to one of 256 patterns, so we measured the illuminance value of each sensor in terms of 2048 patterns (= 256×8 lights). As the energy consumption of each light has a linear relationship to its output, we defined the energy consumption W_i of each light using a calibration curve.The target illuminance values were set as follows:

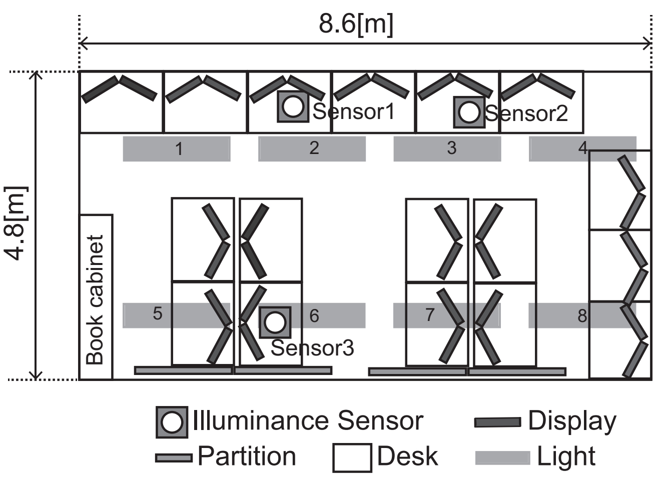


Fig. 2. Experimental environment.

Case1: Sensor1 has a target illuminance of 400 [lx], Sensor2 has a target illuminance of 600 [lx], and Sensor3 has a target illuminance of 500 [lx].

Case2: For the first 50 generations, the target illuminance values are the same in Case 1. After 50 generations, the target value of Sensor 1 increases from 400 [lx] to 600 [lx].

We set β to 1, and the number of loops for tracing E to 1000.

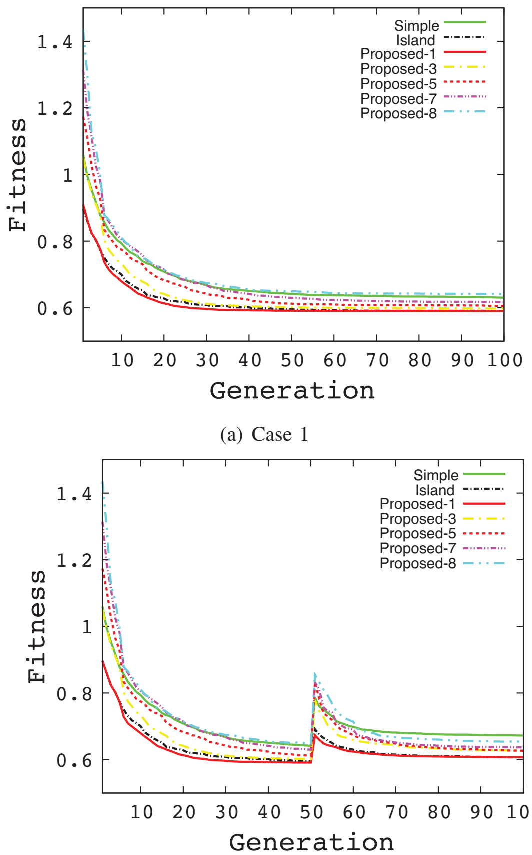
The following standard parameter settings were applied [16], [17], [18]: a recombination rate of 1.0; mutation rate of 0.0; random selection (non-elitist); total population size of $N = 100$; $M = 10$ subpopulations; maximum depth of 17; $I = 100$ generations; migration interval of 5; and initial individuals created by using "ramped half-and-half" with maximum depth.

All experiments were performed on a single PC with six Intel Xeon E5-1660 3.3 GHz processors, 24 GB of memory, and NVIDIA Quadro K6000 running under Linux. The NVIDIA Quadro K6000 is capable of around 5.2 TFlops and 2880 CUDA cores.

C. Performance Evaluation

We first examined whether the GP approach for DLC performed well in terms of satisfying the required sensor values. We conducted 20 trials, and evaluated the best individuals x^u_f in P^u, i.e., $f(x^u_f) \geq f(x)$ $(\forall x \in V^u)$. Figure 6 shows the history of sensor values of x^u_f, where the lines indicate the average over ten subpopulations. Figures 6 (a)–(c) show the results for Case 1, in which the required illuminance values do not change during the search. We confirmed that all GP approaches can satisfy each required illuminance value. These results verify that GP is an effective means of solving the DLC problem. Moreover, Figures 6 (d)–(f) show the results for Case 2, in which the required illuminance of sensor 1 changes after 50 generations. We can see that the illuminance values become unstable around the 50th generation. In particular, the values of sensor 1 vary because the required illuminance changed from 400 [lx] to 600 [lx]. The results show that the required illuminance was achieved after the 50th generation, and that the proposed functions and terminal nodes perform well in both the stable and dynamic environments.

Next, we compared the proposed method with the Simple and Island methods in terms of fitness. Figure 3 shows a plot of $f(x^u_f)$, where the lines indicate the average of $f(x^u_f)$ over ten subpopulations. Figure 3 (a) presents the results for Case 1, and (b) shows the results for Case 2. From Figure 3, it is apparent that the Simple method was outperformed by the other methods in both cases. This implies that migration is necessary in subpopulation models. Although Proposed-1 performed slightly better than the Island method, the other Proposed-α methods gave inferior solutions. This indicates the importance of setting the weight parameter α appropriately. In Figure 3 (b), the required illuminance is higher than before, so the total fitness is high. However, the optimizer of the proposed method converged to a local minimum after the required illuminance changed. Figures 6 (d)–(f) indicate that the required illuminance values were satisfied, demonstrating the effectiveness of the proposed method. Moreover, we verified the validity of the output values $\boldsymbol{L}$ assigned to the lights. Figure 4 shows the average output value given by the Proposed-1 method over 50 trials for Case 1. In this figure, red font indicates the percentage output of the lights. The maximum output value occurs in light 3, which is near the sensor that required the highest illuminance, and the output value of lights 1, 5, 7, and 9, which are far from the sensors, were low. These results validate the optimal output values given by the proposed method.



(a) Case 1

(b) Case 2

Fig. 3. History of fitness.

Finally, we investigated the difference between the proposed method and the Island and Simple methods in terms of energy consumption. Figure 5 shows the average of the history of energy consumption over 20 trials. The best individual x_f^u of subpopulation V^u has the set of output values $\boldsymbol{L}$, and energy consumption is related to $\boldsymbol{L}$. The lines indicate the average energy consumption over 20 trials. From Figure 5,

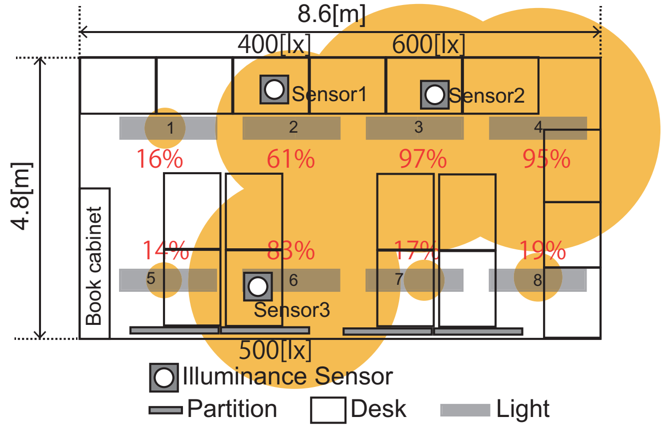


Fig. 4. Output values of lights given by the final generation of the Proposed-1(Case 1).

we can see that Proposed-1 and the Island method outperform the other methods, and the Proposed-α methods are not as stable as the Simple and Island methods. In the proposed method, the migration topology is not fixed, and the number of individuals in each subpopulation can vary. Moreover, the subpopulations are organized based on clustering the features of the individuals, so similar individuals tend to gather in the same subpopulation. This mechanism results in genetic diversity, because subpopulations with different features are generated. Because of this mechanism, individuals with different features are selected as elite individuals, causing the energy consumption to change with time. In this paper, we did not use elite GP strategies. If we had used such strategies, the individuals selected as the elite would not have changed frequently, and neither the energy consumption nor the sensor values would be stable.

D. Behavior Analysis

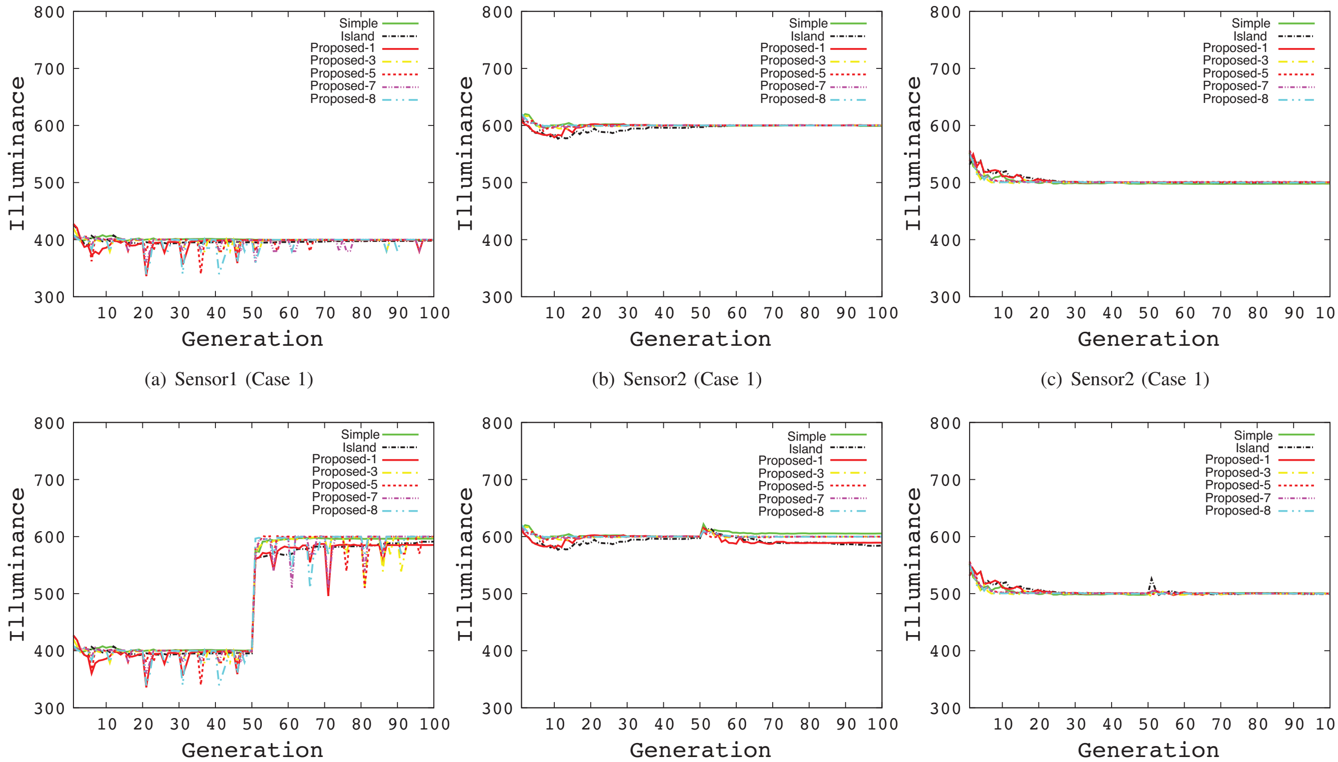
The proposed methods generate a similarity weighted network based on the features of each individual and create an appropriate topology by network clustering. We hypothesized that this mechanism would enhance the genetic diversity. First, we investigated the diversity of output values of lights in comparison with the Proposed-1 and the Island method.

Each individual x_i has terminal nodes and function nodes described in Tables I and II, and the set of output values $\boldsymbol{L}$ is determined by tracing the tree of each individual. We define the set of output values of lights in terms of the best individual x_f^u of subpopulation u by

$$\boldsymbol{L}(x_f^u) = \{L_0(x_f^u), L_1(x_f^u), \ldots, L_I(x_f^u)\}.$$

To investigate the diversity of $\boldsymbol{L}(x_f^u)$, we evaluated the set of averages of $\overline{\boldsymbol{L}(x_f^u)}$, given by

$$\overline{L_i(x_f^u)} = \sum_{u=1}^{U} L_i(x_f^u)/|U|,$$



(a) Sensor1 (Case 1) (b) Sensor2 (Case 1) (c) Sensor2 (Case 1)

(d) Sensor1 (Case 2) (e) Sensor2 (Case 2) (f) Sensor3 (Case 2)

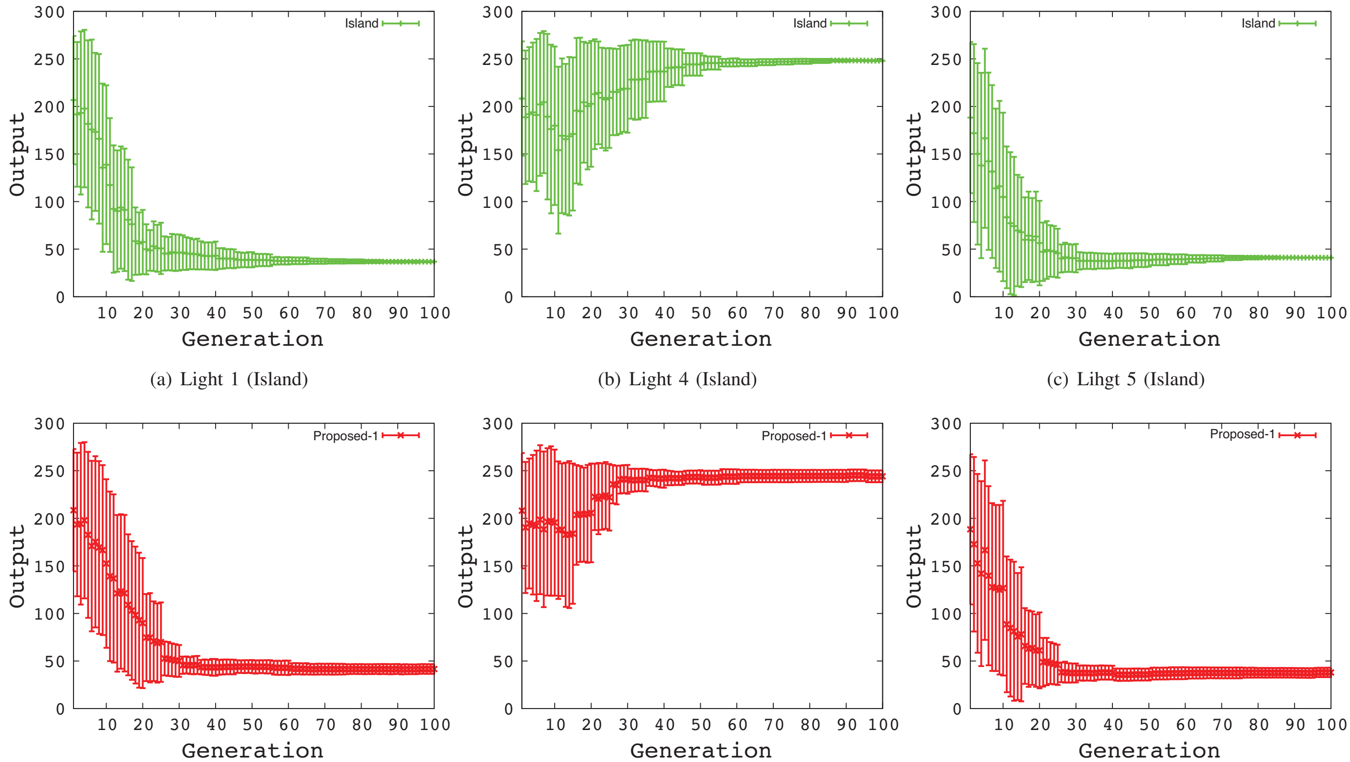
Fig. 6. Comparison of sensor values under different methods.

and the standard deviation of $L_i(x_f^u)$. Figure 7 plots $\overline{L_i(x_f^u)}$ with error bars (showing the standard deviation of $L_i(x_f^u)$). Figure 7 (a)–(c) and (d)–(f) show the results of the Island and Proposed-1 methods, respectively, for Light1, 4, and 5 in Case 1. Due to space limitations, we only show the results for Light1, 4, and 5; however, similar results were obtained for the other lights and with the Proposed-3,-5,-7,and -9 methods. Figure 7 shows that each output value of lights steadily converged, with the standard deviations decreasing with the number of generations. The standard deviation of Proposed-1 is greater than that of the Island method in the latter half of the experiment. In particular, the standard deviation of the Island method after100 generations is much smaller than that of the Proposed-1 method. These results imply that the Island method repeatedly migrates elite individuals, and evolves other individuals using these elite individuals. Thus, the individuals in the Island method become increasingly similar as the search progresses. In the proposed method, instead of a common migration strategy, the topology is adaptively reconstructed. This leads to individuals with similar features gathering within the same subpopulation, which enhances local search and generates subpopulations with different features to enhance genetic diversity. We also verified that the proposed method can encourage genetic diversity in the DLC problem. These results support our hypothesis.

Next, we investigated the length of the individuals. If an individual becomes too long, it takes a lot of time to evaluate its fitness. Because of this “bloat” phenomenon, it is important to analyze the average length of individuals when using the proposed method in real-world optimization problems. Figure 8 shows the average length of x_f^u over 20 trials. Figure 8 (a) shows the results for Case 1, and (b) shows the results for Case 2. From Figure 8 (a), it is clear that the average length increases with the number of generations, but this growth is not exponential. In Case 2, the length decreases immediately after the target illuminance value changed, and the growth with the number of generations is again less than exponential. A similar trend can be observed in all of the comparative methods.

VI. Conclusion

In recent years, it has been reported that intellectual productivity often increases when a user’s brightness requirements are satisfied. Moreover, individual control is expected to reduce the energy consumed by lighting appliances, which accounts for more than 20% of energy consumption in office buildings. To control lights individually, an algorithm must assign a different output value to each light. However, It is considered that this problem has a number of local minima, so an effective algorithm to avoid premature convergence is desired.



(a) Light 1 (Island) (b) Light 4 (Island) (c) Lihgt 5 (Island)

(d) Light 1 (Proposed-1) (e) Light 4 (Proposed-1) (f) Light 5 (Proposed-1)

Fig. 7. Comparison of Proposed-1 and Island methods in terms of genetic diversity.

In this paper, we have defined a fitness function for the DLC problem, and proposed terminal nodes and functional nodes to implement a GP approach. Moreover, we generated an illuminance simulator to evaluate the individuals generated by GP. We evaluated both static and dynamic illuminance scenarios using our GP approach to the DLC problem. To analyze the effectiveness of the proposed method, we introduced five comparison methods that change the optimal balance of the proposed method between the required illuminance values and the energy consupmption.

The experimental results show that the GP approach performs well in solving the DLC problem with the Proposed-1 method exhibiting comparable performance to the Island methods. Moreover, Proposed-1 offers enhanced genetic diversity over common subpopulation methods such as the Island model. These results imply that our CUDA-based Self Adaptive Subpopulation Model in GP (cuSASGP) is applicable to real-world problems. Note that, although this paper has only presented results for two situations, we have achieved similar results in a variety of scenarios.

VII. Future Work

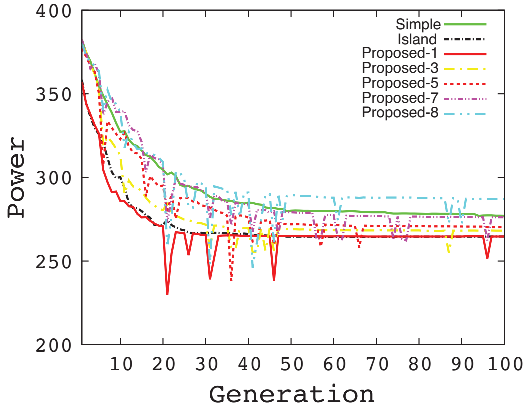
To analyze the search performance of the proposed methods simply, this study did not use elite GP strategies. Our perspectives for future work include the application of elite GP strategies to improve the stability of output values. Moreover, the Building Hypothesis [19], [20], which is the most basic hypothesis in GP, indicates that partial solutions called building blocks are assembled into the entire solution. Thus far, we have accounted for the fitness and node size in cuSASGP as a first step; however, incorporating partial solutions in cuSASGP would lead to more accurate subpopulation reconstruction. The partial solutions often mean that subtree patterns that frequently appear among elite individuals, and the subtree patterns have various size. Therefore, extracting partial solutions generally requires computational effort to enumerate these subtree patterns. In cuSASGP, we have already utilized CUDA for the subpopulation reconstruction. As another area of future work, we intend to construct an effective optimizer for the DLC problem that incorporates partial solutions in cuSASGP using CUDA.

Acknowledgment

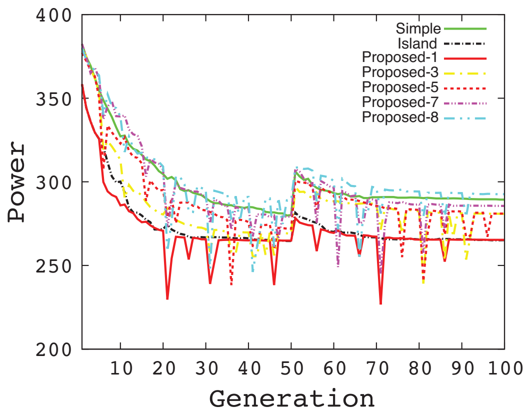
This work was supported by JSPS KAKENHI Grant Numbers 26730133 and 26330290.

References

[1] P. R. Boycea, N. H. Eklunda, and S. N. Simpsona, “ndividual lighting control: Task performance, mood, and illuminance,” *Journal of the Illuminating Engineering Society*, vol. 29, no. 1, pp. 133–142, 2000.

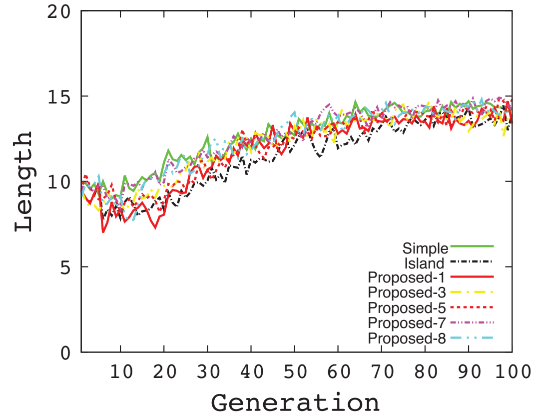


(a) Case 1

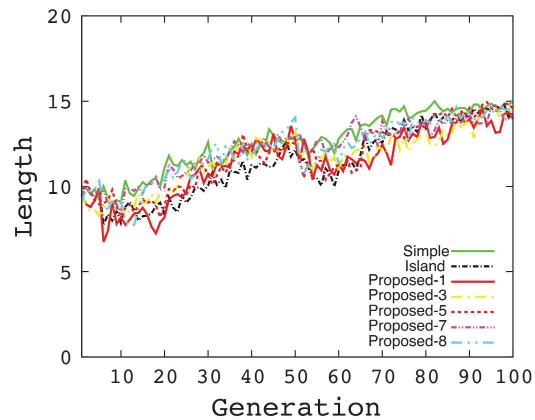


(b) Case 2

Fig. 5. History of energy consumption.



(a) Case 1



(b) Case 2

Fig. 8. Change in average length of individuals.

[2] O. Seppänen and W. J. Fisk, “A model to estimate the cost effectiveness of indoor environment improvements in office work,” *ASHRAE Transactions*, vol. 111, pp. 663–679, 2005.

[3] M. Braun, O. Stefani, A. Pross, M. Bues, and D. Spath, *Human Factors in Lighting*, B.-T. Karsh, Ed. Springer Berlin Heidelberg, 2009, vol. 5624.

[4] F. Rubinstein, “Photoelectric control of equi-illumination lighting systems,” *Energy and Buildings*, vol. 6, no. 1084, pp. 141–150, 1984.

[5] F. Rubinstein, G. Ward, and R. Verderber, “Improving the performance of photo-electrically controlled lighting systems,” *Journal of the Illuminating Engineering Society*, pp. 70–94, 1998.

[6] A. Bierman and K. Conway, “Characterizing daylight photosensor system performance to help overcome market barriers,” *Journal of the Illuminating Engineering Society*, vol. 29, no. 1, pp. 101–115, 2000.

[7] D. Maniccia, B. Rutledge, M. S. Rea, and W. Morrow, “Occupant use of man- ual lighting controls in private offices,” *Illuminating Engineer ing Society*, vol. 28, no. 2, pp. 42–56, 1999.

[8] K. Ono, M. Miki, M. Yoshimi, T. Nishimoto, T. Omi, H. Adachi, M. Akita, and Y. Kasahara, “Development of the intelligent lighting system using led ceiling lights into an actual office,” *IEEJ Transactions on Fundamentals and Materials*, vol. 131, no. 10, pp. 321–327., 2011.

[9] J. R. Koza, *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Norwell, MA, USA: Kluwer Academic Publishers, 2003.

[10] Y. Shichel, E. Ziserman, and M. Sipper, “Gp-robocode: Using genetic programming to evolve robocode players,” in *PROCEEDINGS OF 8TH EUROPEAN CONFERENCE ON GENETIC PROGRAMMING (EUROGP2005)*. SpringerVerlag, 2005, pp. 143–154.

[11] P. Bremner, M. Samie, G. Dragffy, A. G. Pipe, and Y. Liu, “Evolving cell array configurations using cgp,” *Proceedings of the European Conference on Genetic Programming - EuroGP*, pp. 73–84, 2011.

[12] K. Ono and Y. Hanada, “A Cuda-based Self-Adaptive Subpopulation Model in Genetic Programming: cuSASGP.” in *CEC*. IEEE, 2015, pp. 1543–1550.

[13] M. Miki, T. Hiroyasu, and K. Imazato, “Proposal for an intelligent lighting system, and verification of control method effectiveness,” pp. 520–525, 2004.

[14] A. Pandharipande, A. Peruffo, D. Caicedo, and L. Schenato, “Lighting control with distributed wireless sensing and actuation for daylight and occupancy adaptation,” *Energy and Buildings*, vol. 97, pp. 13–20, 2015.

[15] A. E. din Mady, M. Boubekeur, and G. Provan, “Optimised embedded distributed controller for automated lighting systems,” 2010.

[16] K. Yanai and H. Iba, “Estimation of distribution programming based on bayesian network,” in *Proceedings of the 2003 Congress on Evolutionary Computation CEC’03*, vol. 3, dec. 2003, pp. 1618–1625.

[17] W. Langdon and R. Poli, “Why ants are hard,” in *Proceedings of the Third Annual Conference Genetic Programming GP’98*, 22-25 1998, pp. 193–201.

[18] W. B. Langdon and R. Poli, “Foundations of genetic programming,” vol. I-XV, pp. 1–260, 2002.

[19] R. Poli and N. F. McPhee, “General schema theory for genetic programming with subtree-swapping crossover: Part II,” *Evolutionary Computation*, vol. 11, no. 2, pp. 169–206, May 2003.

[20] N. F. McPhee, B. Ohs, and T. Hutchison, “Semantic building blocks in genetic programming,” in *Proceedings of the 11th European conference on Genetic programming EuroGP’08*, 2008, pp. 134–145.