Proposal of Serially and Dynamically Separating Genetic Algorithm and Its Application to Optimization of Robot Control Systems

Koichi NAKAYAMA[1),2)] Hirokazu MATSUI[3)] Katsunori SHIMOHARA[2)] Osamu KATAI[1)]
1) Graduate School of Informatics, Kyoto Univ.
2) ATR Network Informatics Lab. 3) Faculty of Engeenering, Mie Univ.
Address 2-2-2 Hikari-Dai, Keihanna Science City, Kyoto 619-0288, Japan
voice: [+81](0774)95-1098; e-mail: knakayama@atr.jp

Abstract

In this paper, we propose a Serially and Dynamically Separating Genetic Algorithm (sDS-GA), and apply it to optimize an agent-oriented control system for an intelligent robot. The conventional DS-GA is inapplicable as a learning algorithm for a single hardware unit such as an intelligent robot. By the extension of a dynamically separating mechanism, the proposed sDS-GA becomes applicable as an optimization algorithm for a single hardware unit. We conducted experiments with sDS-GA that optimize the parameters of a control system for an intelligent robot called MieC. The sDS-GA obtained not short-term but long-term optimality. We also found that sDS-GA is efficient for the optimization of an actual intelligent robot under an unknown and dynamic environment.

1 Introduction

There have been many studies of control systems for robots[1, 2]. In particular, many intelligent robots operate by agent-oriented programming. However, in agent-oriented programming, it is difficult to design the entire agent optimally beforehand. In this paper, we focus on a Dynamically Separating Genetic Algorithm (DS-GA)[3, 4] as an optimization algorithm for agents in an intelligent robot that is controlled by agent-oriented programming.

An intelligent robot must be designed taking the following into account. It has to do various processing at the same time, such as target determination, image processing, data transferring, and arm and wheel control. The priority of the processing changes in response to the influences of external factors and internal factors, such as interaction with humans, and the residual quantity of the battery. The optimal parameters for processing are dynamically changed by these factors. As such, an adaptation algorithm is required.

In the environment of an actual robot, there is a time lag between the decision of the agent's action and the action of the robot. Whether or not the action is appropriate is determined after a certain time's passing. In order to optimize the control system of an actual robot, because of the above environment, an algorithm that can obtain not short-term but long-term optimality is required.

DS-GA has the ability to increase system-level optimality by the autonomous learning of agents based on local information by using the dynamic separation of the agent's interaction. In other words, system-level information emerges from collective agent-level information by "Swarm-Sensing," which is a characteristic of DS-GA. We expect that extended DS-GA will have the ability to increase long-term optimality by the autonomous learning of agents based on short-term information.

The conventional DS-GA uses the dynamically separating mechanism of its agents. However, in the case where an agent controls a single hardware unit such as an intelligent robot, the DS-GA is inapplicable as a learning algorithm for a robot. In order to optimize a single hardware unit that is designed by an agent-oriented control system such as an intelligent robot, we extend DS-GA to form a Serially and Dynamically Separating Genetic Algorithm (sDS-GA) that includes the time-separating mechanism of the agents.

In DS-GA, many agents act simultaneously, and the interactions of many agents are restricted in a colony. In sDS-GA, a control-agent is chosen serially and the influence of the agent's action is decreased with time. In order to reduce an interaction with an agent that belongs to other colonies, the agents who belong to the same colony are chosen continuously.

In order to verify the validity of the proposed sDS-GA, we applied the sDS-GA to an object tracing task for a Movable Intelligent Evolutional Computer (MieC) as an actual robot.

2 Serially and Dynamically Separating Genetic Algorithm (sDS-GA)

In this section, we propose the Serially and Dynamically Separating Genetic Algorithm (sDS-GA) as an applicable DS-GA for an actual robot. The conventional DS-GA uses dynamic separation as follows. Agents that are separated into colonies act simultaneously. The interactions of agents are restricted in a colony, and an agent cannot contact any agent that belongs to another colony. The colonies change dynamically according to the number of agents they contain. When the number of agents in a colony increases, the colony is divided into two halves. A colony is extinguished when the number of agents it contains becomes 0.

We propose sDS-GA for use as a control system for an actual robot. The basic idea of sDS-GA is as follows. In sDS-GA, control-agents are separated into colonies. Agents in a colony control a robot in a period serially. In a certain period, control-agents who belong to a certain colony are chosen randomly, and they control the robot serially. In the next period, control-agents who belong to the next colony are chosen randomly, and they control the robot serially.

As a result, the interval of control by an agent that belongs to the same colony is much shorter than the interval of control by an agent that belongs to other colonies. In other words, the influence by a certain agent's action is strong for the agent that belongs to the same colony, and weak for the agent that belongs to other colonies. Serial separation is realized by such a mechanism. We show the main routine of the algorithm using sDS-GA in **Fig. 1**.

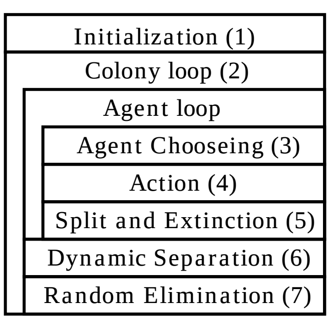

Figure 1: Main routine of the sDS-GA shown by NS chart.

Specifically, in the experiments described in this paper, the evolution of a population is based on the split or extinction of agents according to their private performance, e.g., accumulated profit. Consequently, an agent's autonomy is not spoiled and agents can still learn by means of evolution. The learning algorithm used by the DS-GA is as follows.

(1) Initialization: $N_A(t)$ agents at $t = 0$ are created and separated into colonies. The number of agents in a colony is $N_{Lim}/2$. The evaluation value of an agent a, $E_A(a,t)$, is initially set to $E_A(a,0)$ and its action determination gene, $Gene_{Act}(a)$, is initially randomly chosen.

(2) Colony Loop: Every colony takes charge of the control in order.

(3) Agent Choosing: An agent is chosen from the colony for robot control randomly.

(4) Action (Robot control of agent): The agent acts for robot control. Details are shown in section 3. The agent changes its own evaluation value based on the result of the action.

(5) Split and Extinction of Agents: An agent is split into two agents when the evaluation value by an agent becomes more than twice the initial value ($E_A(a,0)$). The two agents inherit half of the original agent's evaluation value. An action gene is mutated according to the mutation probability P_{mut}. An agent is extinguished from the colony when its evaluation value becomes less than or equal to zero.

(6) Dynamic Separation of Colonies: When the number of agents in a colony exceeds the limit N_{Lim}, the agents are separated into two half-colonies. The difference in the number of agents between the two colonies is either 1 or 0.

(7) Random Elimination: When the total number of agents that can exist in the robot control system becomes greater than the initial number of agents (N_{Lim}), a colony is eliminated at random.

In the experiments, the number of initial agents was set to $N_A(0) = 100$, the number of maximum agents in a colony was set to $N_{Lim} = 20$, the mutation probability $P_{mut} = 0.1$, and the initial accumulated profit $E_A(a,0) = 100$ for all agents.

3 Experiment

Each agent has an evaluation value for its own task achievement, and has no information about the other task achievements. We think that the total of their evaluation values is maximized, so that the system becomes optimal. But, it is not always maximized as an entire system, even if each agent acts in order to maximize its task achievement, i.e., the system may have a dilemma on a task achievement such as the following. An agent may be unable to use computer resources to do a task, if another is doing a task by using computer resources. An agent may be unable to maximize a task achievement in a longer term, if it maximizes a task achievement in a shorter term.

In this section, we verify experimentally that sDS-GA is efficient for actual intelligent robots. Here, we use an auto guided tracked vehicle with a camera, called (MieC) as an actual intelligent robot.

Concretely, applying sDS-GA, the control parameters of MieC are optimized when MieC traces a ball that moves on the same plane.

3.1 Movable Intelligent Evolutional Computer (MieC)

Here, we use the Movable Intelligent Evolutional Computer (MieC) shown in **Fig. 2**as an actual intelligent robot. MieC has two motors as movable actuators, two encoders for the motors as internal sensors, and a camera as an external sensor.

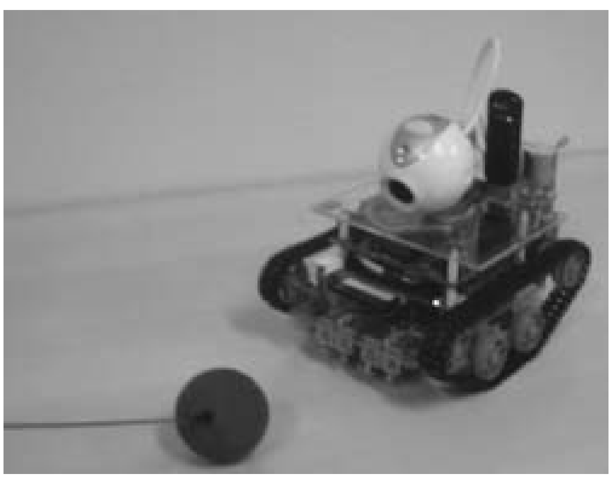

Figure 2: Movable Intelligent Evolutional Computer (MieC)

3.2 Tracing Control

The flow of target tracing control is shown in **Fig. 3**. This control flow is one of the simplest in this case. (1) The current position of the target is solved by processing an image from the camera. (2) The position command of MieC is determined by the error between the target position and the current position. (3) The target rotation angles of the left and the right tracks are calculated from the position command, considering the inverse kinematic of MieC. (4) The voltage commands to motors are determined by the errors between the target positions and the value current positions, respectively. (5) The motors are rotated by the voltage commands, respectively. (6) Return to (1).

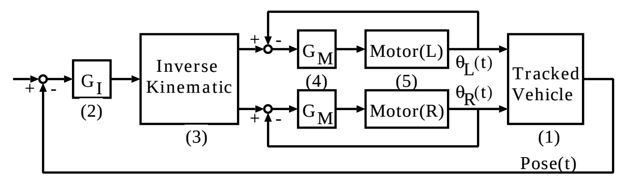

Figure 3: The block diagram for the target tracing of the traced vehicle

In steps (2) and (4), the commands are determined by the errors and the gains G_I and G_M. Here we treat the gains G_I and G_M as proportionality constants, although the gains have to be determined, considering the weight and the inertia moment of MieC and the characteristics of the motors, etc. The execution cycle (1)$\sim$(3) and the execution cycle (4)$\sim$(5) depend on the execution speeds of image-processing and motor-control, respectively. These execution cycles are considered to determine the gains G_I,G_M. Here, these parameters G_I,G_M are optimized applying sDS-GA.

3.3 sDS-GA Coding

We give each agent gene G_I or G_M. G_I takes one of 10 quantized values 0.0, 0.1, 0.2, $\cdots$, 0.9. G_M takes one of 10 quantized values 0.0, 0.3, 0.6, $\cdots$, 2.7. The agents with gene G_I and gene G_M can call the cycle(1)$\sim$(3) and the cycle(4)$\sim$(5), respectively. However, the agent cannot call a cycle before the same cycle is completed. Each agent is evaluated by how much and how fast MieC can trace the target during the time in its duty.

3.4 Experiments and Discussion

We gained the following results. The histories of the population ratios of G_I(0.0, 0.1, 0.2, $\cdots$, 0.9) and G_M(0.0, 0.3, 0.6, $\cdots$, 2.7) are shown in **Fig. 4**, respectively. The horizontal axes express the number of times the colony loop was performed. The vertical axes express the ratios of G_I and G_M with each of 10 quantized values, respectively.

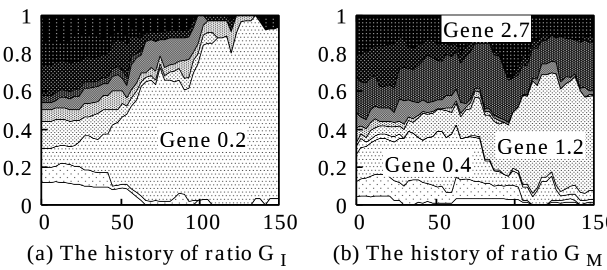

Figure 4: The history of the population ratio

G_I converged on the optimal value 0.2. G_M did not converge, but the average of G_M in each colony converged near the optimal value 1.8. It is believed that the action time of each agent is so short in this case that an agent's action affects not itself but other agents in the same colony, evenly. In other words, agents in the same colony evenly affect each other. As the results, the average in each colony converged, but the agents did not converge. Therefore, a single agent with G_M cannot aquire the optimal value, but aquires the optimal value as the average. On the other hand, a single agent with G_I aquired the optimal value. It is believed that the average was equal to the value of the agent, since only a few agents with G_I are in the same colony.

From the above discussion, it is believed that both G_I and G_M are optimized by applying sDS-GA. The sDS-GA is efficient for the optimization of a control system for MieC as an actual intelligent robot under an unknown environment.

4 Conclusion

In this paper, we proposed the Serially and Dynamically Separating Genetic Algorithm (sDS-GA) as an optimization algorithm for an actual intelligent robot control system.

In order to verify the validity of sDS-GA for the optimization of intelligent robot control, we applied sDS-GA to a target tracing task for a Movable Intelligent Evolutional Computer (MieC) as an actual intelligent robot.

Experimental results show that even if there is a time lag between the decision of the agent's action and the action of the robot, and even if the agent learns based on short-term information, sDS-GA can obtain not short-term but long-term optimal parameters of the control system. These characteristics of sDS-GA are similar to the characteristics of the conventional DS-GA, which obtains system-level optimality by agents' learning based on local information by the use of "Swarm-Sensing". These results suggest that sDS-GA is efficient for the optimization of an actual robot control system under an unknown environment.

Acknowledgements

This research was conducted as part of 'Research on Human Communication' with funding from the National Institute of Information and Communications Technology.

References

[1] A. Mori, F. Naya, N. Osato, T. Kawaoka, "Multiagent-Based Distributed Manipulator Control," *Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 289-296, 1996.

[2] J.D. Morrow, "Sensorimotor Primitives for Programming Robotic Assembly Skills," *Ph.D. Thesis, Robotics Institute, Carnegie Mellon University*, 1997.

[3] K. Nakayama, H. Matsui, Y. Nomura, "Dynamically Separating GA (DS-GA)," *IPSJ Transactions on Mathematical Modeling and Its Applications (In Japanese)* , Vol. 43, No. SIG 10 (TOM 7), pp. 95-109, 2002.

[4] K. Nakayama, K. Shimohara, O. Katai, "Dynamically Separating GA: A New Method of Achieving the System-level Optimality in MAS," *The journal of three dimensional images*, Vol.16, No.4, pp. 177-183, 2002.