Journal of Robotics, Networking and Artificial Life, Vol. 2, No. 2 (September 2015), 107-110

On the Effects of Epigenetic Programming on the Efficiency of Incremental Evolution of the Simulated Khepera Robot

Yasuto Nishiwaki, Naoki Mukosaka, Ivan Tanev, and Katsunori Shimohara
Graduate School of Science and Engineering, Doshisha University,
1-3 Tatara-Miyakotani, Kyotanabe, Kyoto 610-0321, Japan

E-mail: nishiwaki2012@sil.doshishsa.ac.jp, mukosaka2012@sil.doshisha.ac.jp, itanev@mail.doshisha.ac.jp, kshimoha@mail.doshisha.ac.jp

www.doshisha.ac.jp

Abstract

We present the results of our work on epigenetic programming (EP) and its application to navigation of mobile cleaning robot. We verify the effect of genetic switches and the feasibility of incorporating histones. The results show individuals applying Lamarckian EP have better quality and evolve faster than those in canonical genetic programming. Such an improved computational effort of evolution, however, is achieved at the cost of somehow decreased computational performance due to the additional overhead of manipulating the genetic switches. Our work could be seen as a verification of the accelerating effect of histones on adaptation and evolution in nature.

Keywords: Epigenetic Programming, Genetic Programming, Epigenesis, Histone.

1. Introduction

The objective of our study is to develop an approach of Epigenetic Programming[1-2] (EP) and to verify the feasibility of incorporating histones (the family of proteins which DNA is wrapped around) in genotype from both the computationally and natural perspectives. Epigenesis refers to a changeable gene expression (and consequently – a variable phenotype) as a result of the interaction of genotype and surrounding environment (food, stress, toxins, radiation, etc.). We intend to mimic the recent discoveries in molecular biology that suggest that epigenesis plays an important role in evolutionary adaptation of species.

In our previous study[1], we verified the beneficial effects of histones on the quality of solutions obtained via Genetic Programming (GP). Out approach implied that the modifications to histones are implemented *after* the completion of GP, and these modifications were considered as a local search (hill-climbing) in the proximity of the genotypic space of the best-evolved individuals. No modifications to histones in due course of evolution were assumed in our previous work.

Published by Atlantis Press

2. Mechanism of EP

2.1. *The mechanism of modifying the histone switches*

We introduce histones as genetic *on* and *off* "switches" that modify the gene expression by activating and deactivating the nodes (and corresponding sub-trees) in the tree representations (i.e., the "genotype") of the evolved individuals in Genetic Programming (GP). In canonical GP the abilities of individuals evolve as a result of modifications (via crossover and mutation) to the genes alone. In the proposed EP this genetic modifications are superposed with the modifications of histones to facilitate even better evolutionary adaptation to the environment. Moreover, in some cases the modification to histone alone – i.e., by enabling a disabled code (or vice versa) – would achieve a better performing individual without the need to spend extra time for computationally costly evolution of a new genotype. In this study, we use Lamarckian EP (LEP) as elaborated below.

2.2. *The mechanism of LEP*

LEP is an epigenetic approach in which the modified genetic switches (histones) are stored back in the genotype (i.e., inherited). The Lamarckism implies that the modifications to the histones are inherited back into the genotype of the evolving individuals. LEP evaluates the fitness of individual after applying changes to histones, which, in turn, modifies the state of the genetic *on* and *off* switches. In such a way, by changing the expression of the genes (rather than changing the genes) by these switches, it could be possible to obtain a different phenotype, which eventually features a better fitness value of the evolved individuals. Furthermore, we believe that is possible to evolve better individuals as a result of crossover and mutation operations that operate over the space of genotype superposed by histones.

3. System Architecture

In this study, we use the XML-based genetic programming framework[3] (XGP) to maintain the population of evolving individuals and to perform genetic operations (selection, crossover and mutation) on them. The evolved individuals are controllers of a simulated Khepera vacuum cleaning robot. The latter is simulated in a Webots[4] mobile robot simulation environment.

The system architecture is shown in Figure 1.

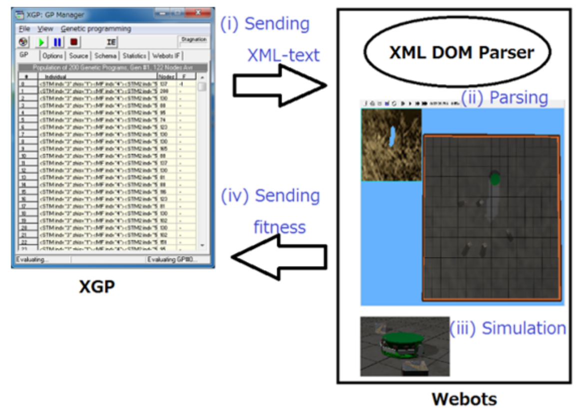

Fig. 1. System architecture

The evaluation of the fitness of the evolved controller is accomplished in following four steps:

(i) XGP sends the genotype of the controller (as XML-text) for evaluation to Webots via UDP channel,

(ii) Using XML parser, Webots parses the received controller into a DOM-tree,

(iii) For each time step of simulation, Webots evaluates the DOM-tree, and based on current perceptions (sensory information) of the simulated bot, decides its current action (e.g., moving forward, moving backward, turning left or turning right). Webots applies the currently issued command on the robot and updates its speed and orientation, which, in turn, defines the position, orientation and the velocity of the bot for the next time step.

(iv) Upon the completion of simulation, Webots returns (via UDP channel) the fitness value (i.e., the area on the floor that was cleaned by the robot) to the XGP.

The controllers of the robot are evaluated for a number of time steps, given to the robot to clean a room, starting from a given starting position in two consecutive sample 2D scenes – (i) without- and (ii) with obstacles, respectively, as shown in Figure 2.

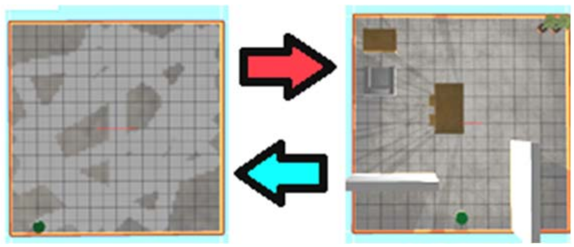

Fig. 2. Experimental setup consisting in two sample scenes: without- (left) and with obstacles (right), respectively. The robot evolves the ability to clean these two rooms consecutively, at two consecutive stages of an incremental evolution.

4. Experiments

To verify the feasibility of incorporating histones, we performed two experiments. These two experiments were intended to allow us to compare the efficiency of evolution via LEP and canonical GP, respectively.

4.1. *Exploring evolution characteristic of LEP*

In this experiment, we explore how histones influence the efficiency of evolution. The controller of the robot is evolved in the first stage (i.e., room without obstacles) of LEP under the same conditions as GP. We believed that introducing histone would facilitate a faster evolution in that it would require lower computational effort (i.e., would feature a reduced number of evolved generations).

4.2. *Two-staged incremental evolution*

During the first stage of the incremental evolution, the robot evolves via GP its ability to efficiently clean the room without obstacles. Then, the best controllers are used to seed the second stage of the evolution which involves an evolution (via GP) of the ability to clean the room with obstacles. The same two-staged incremental evolution is used applying LEP with histone-modification at both stages.

After that, the best controllers of second stage are evaluated again in the room without obstacles (i.e., the same environment that is used during the first stage) to investigate their abilities to adapt to the "new" environment. Because of changing environment, the controllers would need to evolve new features in order to perform well in it. We believe that individual used applying LEP can adapt to the new environment by just switching some of their genes *on* and *off* instead of evolving new genotype from scratch.

5. Experimental Results

5.1. *Result of one-staged evolution*

The results of applying LEP or GP on individuals evolved are shown in Figure 3.

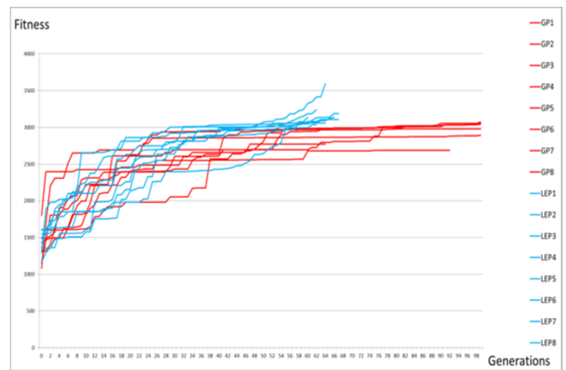

Fig. 3. Fitness convergence of LEP (blue) and GP (red), respectively. The maximum number of generations is 100. The number of evaluations per one generation of individuals applying LEP is two times as many as GP.

As Figure 3 indicates, the controllers, evolved via LEP have better fitness and evolve faster (with lower number of generations) than canonical GP. On the other hand, the average complexity of genotype (in number of nodes in their DOM-trees) is higher in LEP, which, in turn result in an increased computational overhead because of slower evaluation of these genotypes. This finding implies that the improvement of computational effort due to the incorporation of histones in LEP comes at the cost of some degradation of computational performance. This degradation, however, is rather negligible, because, in the considered application of LEP, the runtime, needed to simulate the physics of the Khepera robot is much longer than the time spent for evaluating the DOM-tree of the genotype.

5.2. *Result of two-staged incremental evolution*

In second stage, even if there is no difference in the efficiency of the first stage, performance of controllers applying LEP was better than GP. LEP's individual tended to live but GP's individual easy to be weeded out difficult stage. However, some individuals tended to not adapt in first stage when the best individuals in the second stage are returned.

6. Discussion

The experimental results suggest that introducing histones as genetic switches facilitates a faster evolution of a simulated Khepera robot. The simulated evolution is relatively fast anyway, and, features about the same runtime overhead as the adaptation via changing of histones. In the real biological world however, the evolution of one generation requires years of time, while the effect of modifications to histones could be observed much sooner. Therefore, we believe that histones incorporated in the genotypes are capable of facilitating the biological evolution.

In addition, in latter experiment, we discovered that some of the controllers could not adapt well to the old "new" environment (room without obstacles), because the parts of their respective genotypes that are responsible for dealing with that environment, was lost during the second stage of evolution (in a room with obstacles). Therefore, it could be interesting to investigate such a genetic mechanism during the second stage of LEP that could protect the useful features (e.g., ability to move and clean a room without obstacles) evolved at the first stage of evolution, from the destructive effect of crossover and mutation operations. One possibility is to allow the inactivated histones to repel the crossover and mutation operations, in order to preserve the inactivated code from destruction. This code could be deactivated later, when an old "new" environment arises.

7. Conclusion

We proposed an approach of LEP and verified the feasibility of incorporating histones. In order to implement histone, we introduced genetic "switches" that modify the gene expression by activating and deactivating the nodes in the tree representations (i.e., the "genotype") of the evolved individuals in GP. In our approach we propose LEP in that the modified genetic switches are inherited back in the evolving genotypes.

Also, we verified the effect of epigenesis on the efficiency of evolution of controllers of simulated Khepera cleaning robot in two experiments. From the results of the former experiment we concluded that applying LEP facilitates an earlier adaptation compared to canonical GP. The results of the latter experiment - with two-staged incremental evolution – indicate that using LEP contributes to the efficiency of evolution to old "new" environment. We think these results suggest that histones could be seen as a mechanism to "store" dormant abilities once needed to deal with challenging environments. These abilities could be activated again when the organisms face "new" environments that have been already explored in their evolutionary past.

References

1. Y.Nishiwaki, I.Tanev, and K.Shimohara: Epigenetic Programming and its Application to Mobile Robot, Int. Conf. on Humanized Systems 2013, pp.66-68, September 2013
2. I.Tanev and K.Yuta: Epigenetic programming Genetic programming incorporating epigenetic learning through modification of histones, Information Sciences, p13 (2008)
3. I.Tanev: DOM/XML-based portable genetic representation of the morphology, behavior and communication abilities of evolvable agents, Artificial Life and Robotics, 8:52-56,2004.
4. O.Michel, Webots: Professional mobile robot simulation, Int. J. Adv. Robot. Syst, 1, 39–42 (2004).