

早稲田大学大学院情報生産システム研究科

博 士 論 文 概 要

論 文 題 目

Study on Genetic Network
Programming with Variable Size
Structure and
Genotype/Phenotype Mapping
Mechanism

申 請 者
Bing LI

情報生産システム工学専攻
ニューロコンピューティング研究

2013 年 4 月

Darwin's theory of evolution firstly gives a convincing explanation of the origin of creatures, and shows the fantastic power of nature selection. Inspired by the theory, Evolutionary Algorithms (EAs) are developed by computer scientists, which are kinds of meta-heuristic optimization algorithms for solving the problem with very large search space by improving the candidate solutions iteratively. The essential advantages of EAs are to solve the particular problem with less prior knowledge and human intervention. Starting from the 1950s, a lot of EAs are developed, such as Evolution Strategies (ES), Genetic Algorithm (GA), Genetic Programming (GP) and Evolution Programming (EP). They have been successfully applied to many fields such as engineering, biology, economics, marketing, robotics, physics, chemistry, education and so on.

After investigating the benefits and shortcoming of GA and GP, Genetic Network Programming (GNP) was proposed around 2000. The directed graph structure of GNP extends the chromosome representation of strings in GA and trees in GP, which makes it have high expression ability with relevant small size of individuals, and consequently GNP has the better performance than other evolutionary algorithms. Nowadays, GNP is not only used to solve benchmark problems but also applied to many real world applications such as elevator supervisory control systems, stock market prediction, data mining and traffic prediction.

Since GNP was proposed, many methods have been developed to improve the performance of GNP such as combining GNP with reinforcement learning, introducing symbiotic learning in GNP, upgrading the structure of GNP by defining macro node and rule accumulation. Although these methods have been proved to improve the performance of GNP by combining some other machine learning methods, some useful prior knowledge of biology: increasing length of gene, evolution by gene duplication and genotype-phenotype mapping, are not well considered.

Therefore, the objective of my research is to improve the performance including the expression and generalization ability of GNP by applying the above theories and upgrading the structure of GNP. And, in this research, two kind of methods and their extensions have been proposed to achieve this objective.

Chapter 1 firstly gives the background about the history of Evolutionary Algorithms and some related biology theories, then describes the concept of Genetic Network Programming, finally proposes the objective of my research.

Chapter 2 proposes a new type of GNP -- Variable Size Genetic Network Programming (GNPvs), which inspired by the increasing length of gene in species. In this method, the size of the individuals in GNPvs could be changed by evolution by a new crossover developed to replace the uniform crossover in GNP. The new crossover makes some nodes to move from one parent GNP to another parent GNP following binomial probability distribution. In the simulations of the tileworld, the effectiveness of the proposed method is verified by comparing with GNP. Through examining different crossover rate, initial size and ratio of judgment nodes and processing nodes of GNP and GNPvs, it is confirmed that the proposed method can obtain the optimal size and optimal ratio of judgment nodes and processing nodes of individuals, which results in improving the effectiveness of GNP.

Chapter 3 develop a kind of replacement mechanism by learning the concept of build block hypothesis and evolution by gene duplication, in order to improve the generalization ability of GNPvs, The extension method of GNPvs is named GNPvs with Replacement (GNPvs-R), in which the sets of frequently used nodes are extracted from elite individuals and these blocks are used to replace the non-frequently used nodes of individuals. With the help of this mechanism, the whole structure of the individual will be evolved and the most valuable information from elite individuals will be contributed to all individuals to the population. In order to test the generalization ability of the proposed method, the randomly generated tileworld is used in the simulations, i.e., the location of obstacle, tiles and holes is randomly created when it is initialized. After simulating the proposed method on 2 training worlds and 8 testing worlds, it is found that the proposed method increases the generalization ability of GNPvs.

Chapter 4 tries to modify the structure of GNP by using the genotype phenotype mapping mechanism to cope with automatic program generation task. The proposed method is called GNP-APGm, in which the individual is a solution generator with outside memory. The mapping mechanism allows the proposed method to create programs by the communication between GNP

structure and outside memory. Moreover, the proposed method develops two functions "IF" and "SUB2" to create two basic statements in a program, i.e., conditional statements and sequential statements. Through the transition of nodes and the mapping process, it does create not only simple statements, but also some complex programs to deal with the problem. The proposed method has advantages of using graph structures fully, keeping the diversity of the genotype and using the building blocks and subroutines, which is benefitted to find better solutions. In the simulations, the performance of GNP-APGm is firstly evaluated by comparing with GNP using a number of small tileworlds, then the parameters of GNP-APGm, i.e., the number of transitions, the maximum length of program and the number of the subprograms, are discussed. It is confirmed from the simulation results that the proposed method is more robust than GNP, since it can generate better solutions than GNP.

Chapter 5 introduces the three steps problem solving methodology in GNP-APGm, i.e., decompose a complex problem into some simpler problems, then finds solution to cope with each subproblem, finally seeks a way to assemble these solutions of the subproblems into one solution for the original problem. The proposed method is named Subroutine embedded GNP-APGm (GNPsr-APGm), in which plural GNP structures are divided into two parts, i.e., main part for generating main function and subroutine part for creating subroutines. Since the program is decomposed into a main function and subroutines, each part of the program can deal with one specific subproblem, which makes the proposed method find a better solution more effectively and efficiently. Moreover, the size of the program is decreased by reusing the subroutines obtained by the proposed method. The performance of GNPsr-APGm is firstly evaluated on Artificial Ant problem to affirm whether the proposed method can generate useful subroutines. Then, the proposed is tested on the tileworld to verify the improvement compared with GNP-APGm. It is confirmed from the simulation results that the proposed method can generate hierarchical programs efficiently and improve the performance of GNP-APGm.

Chapter 6 makes conclusions of this thesis by summarizing the contributions of this research.