International Journal of Hybrid Information Technology
Vol.8, No.9 (2015), pp.95-104
http://dx.doi.org/10.14257/ijhit.2015.8.9.11

Software Testing Case Generation of Ant Colony Optimization Based on Quantum Dynamic Self-Adaptation

Chen Ping and Xia Min

Ma'anshan Teacher's College, Ma'anshan, 243041
anhuicp@126.com
Anhui University of Technology, Ma'anshan,243002
2237690981@qq.com

Abstract

Automatic generation of testing data is the most crucial technology in the testing phase with some certain real value to improve software's testing automation degree. Although some certain results have been achieved by introducing the ant colony optimization to the testing process, the algorithm itself has the defects like too fast convergence speed and being easy to fall into local optimum. Based on this, self-adaptive factor is introduced in this paper to balance the algorithm's local optimum. Meanwhile, quantum algorithm is introduced to realize the individual renewal of the ant colony and reduce the ant colony's size so as to further optimize the target problem. Simulation experiments show that in the multipath test, algorithm in this paper has achieved remarkable progress compared with the ant colony optimization and accuracy of automatic data's optimal solution has been improved.

Keywords: *Quantum Algorithm, Self-Adaptive Factor, Ant Colony Optimization*

1. Introduction

Software testing, which has always been very important content in software engineering, is mainly to be able to find and correct errors in the software system so as to improve the reliability of software. It can be found through previous project development experience that software testing spending accounts from 40% to 50%, and the automatic case in software path testing has great significance to improve the efficiency of software testing and reduce software testing costs [1]. Scholars both at home and from abroad have conducted some research about the testing and introduced intelligent algorithm to the present testing. Literature [2] begins to apply genetic algorithm to the automatic generation of testing data; Literature [3] proposes the function minimization method. Fu Bo [4] combines the simulated annealing algorithm with genetic algorithm and has obtained certain achievements; Li Jun [5] and other people propose the self-adaptive genetic algorithm and improve the algorithm's coverage rate through changing the crossover rate and mutation rate by self-adaption; Xia Yun [6] and other people make some improve to the genetic algorithm and propose a method to generate testing data automatically based on IGA, effectively improving the defect of premature convergence of genetic algorithm. Due to the complex algorithm and parameter setting of genetic algorithm, Li Aiguo [7] and other people propose a software structure testing data

ISSN: 1738-9968 IJHIT

automatic generation method based on particle swarm algorithm.

In recent years, the optimization of ant colony optimization has become the focus of research as a testing case data production algorithm with low complexity and good robustness and the characteristics of good global search ability. Its shortcoming is that it does not have strong local-space searching ability and being easy to fall into local optimum and be premature. Aiming at the shortcomings of ant colony optimization, quantum algorithm and self-adaptive factor are introduced in this paper so as to make the improved ant colony optimization has small colony size as well as superb performance and the abilities of fast convergence speed and global optimization at the same time so that it can quickly produce testing cases. In this paper, the improved ant colony optimization of improved quantum dynamic self-adaption is used in the multipath coverage testing case method and improves the effectiveness of algorithm in this paper through experiments.

2. Software Testing Data Production

Software testing is a complete process including many different testing behaviors. Automatic software testing technology is for a continuous and iterative process. Through the research of automatic software testing technology, an appropriate framework model can be achieved so as to make the entire testing process can well combine various auxiliary testing tools. Suppose the input variable for the testing program is $x_1, x_2 \cdots x_n$, and output variable is $y_1, y_2 \cdots y_n$, then functions realized are as follows:

$$V_i = \{(x, y) \mid (x_1, x_2 \cdots x_n) \to (y_1, y_2 \cdots y_n)\} \qquad (1)$$

Herein, each V_i has a corresponding value range D_i, and through the Cartesian product, all the output variables become the domain of definition D and $D = D_1 \times D_2 \cdots D_n$. The control structure of the testing program is represented with diagrams, $G = (V, E, S, e)$, V refers to the corresponding nodes of all the variables in the diagram, E refers to the corresponding set controlling the flow in testing sentences, S is the starting point of testing program and e is the end.

3. Introduction to Ant Colony Optimization

Ant colony optimization, first proposed by the Italian scholar Dorigo M [8] and other scholars in 1992, is a simulation evolutionary algorithm obtained by simulating ants. This algorithm does not rely on the mathematical description of specific issues, thus it has strong ability for global optimization and inherent parallelism, and has achieved good results in many combinatorial optimization problems. When ants go out in search of food, they will release a substance called pheromone on the way, and this substance can guide subsequent ants to choose the way they will go and the subsequent ants will choose the path with high concentration of pheromone. Thus it will lead a certain path will has a number of ants to pass by and it will have more pheromone so that more subsequent ants will choose this path. A positive feedback mechanism will be formed in this way, and finally all the ants will be guided to the shortest path from the nest to the food source.

Suppose m refers to the number of ants, d_{ij} refers to the distance between city i and city j, $\eta_{ij} = 1 / d_{ij}$ is heuristic information, τ_{ij} refers to the amount of pheromone at ij, and

$p_{ij}^{k}(t)$ is the probability of the k ant in city i to choose city j at the moment t. Then:

$$p_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s \in allowedk}[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, j \in allowed_{k} \\ 0, otherwise \end{cases} \quad (2)$$

$allowedk$ is the collection of cities that can be chosen at present

α and β refer to the weight of τ and η in calculation respectively. When α =0, the nearest city is likely to be chosen, and the essence of this algorithm is a greedy algorithm, when β =0, only the pheromone works, it will become a purely positive feedback stochastic heuristic search algorithm.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \quad (3)$$

In the formula (3), ρ refers to the pheromone evaporation factor, and $\Delta\tau_{ij}^{k}(t)$ refers to the pheromones released by the k ant when it passes the path (i, j) at the moment t.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L_{k}}, Ant\ in\ a\ loop\ through\ the\ first\ (i, j) \\ 0, otherwise \end{cases} \quad (4)$$

It can be found in formula (4) that L_{k} refers to the pheromone and the length of the route of ant k at the moment t. The amount of pheromone released by ants on the path they have passed is negatively correlated with the length of the route, and positively correlated with the quality of the route. The route with more pheromone will have more chance to be selected, but with the increase of path, problems will arise like too long searching time and being easy to fall into local optimum, then the optimal solution to the algorithm is easy to be lost.

4. Testing Data Generation of Ant Algorithm Based on Quantum Self-Adaptation

4.1 Self-Adaptive Algorithm

Firstly, though self-adaptive algorithm in the algorithm, it is an effective method to improve the performance by introducing dynamic self-adaptive factors. The main function of dynamic factors is to further search the space. It is easy to fall into local optimum when the searching space of the algorithm in the space is too small, and when the range is too big, it is easy to reduce the convergence speed and become stochastic search. The searching range is adjusted according to the running state of the search. At the beginning of the global information, expand the maximization searching scope as much as possible so that the algorithm can be positioned to a better searching area. In the middle and later phase of the global information, the searching range should be narrowed gradually to ensure the continuation of optimization searching.

Suppose the amount of colonies in ant colony optimization is N , the j individual in the

group i of ants x_i is $x_{ij} = (x_{ij1}, x_{ij2}, \ldots\ldots x_{ijn})$, herein, n is the dimension, the optimal position of the j historical ant is $y_i = (y_{j1}, y_{j2}, \ldots\ldots y_{jn})$. Combine the current position of ant colony with its historical optimal position, and mark it as $z_j = (x_{ij1}, x_{ij2}, \ldots\ldots x_{ijn}, y_{j1}, y_{j2}, \ldots\ldots y_{jn})$, all the individual ants z_j form a $N \times n$ matrix z. After the normalization process, the following matrix z' can be obtained

$$z' = \frac{z_{gl} - \min_{1 \le u \le N, 1 \le v \le n} z_{gl}}{\max_{1 \le u \le N, 1 \le v \le n} z_{gl} - \min_{1 \le u \le N, 1 \le v \le n} z_{gl}} \qquad (5)$$

Each row vector z_j' is a fuzzy set representing the location of destination the j ant reaches and the degree of memberships of each component at historical positions, and the similarity of any two z_i' and z_j' is denoted with similarity

$$Q(u,v) = \frac{\sum_{i=1}^{n} \left| z_u' - z_v' \right|}{n} \qquad (6)$$

The diversity measurement degree F of ant colony is denoted with average degree of closeness

$$F = \frac{2\sum_{n=1}^{N-1} \sum_{i=n+1}^{N} Q(u,v)}{N(N-1)} \qquad (7)$$

According to the above description, the strategy formula at global searching is as follows:

$$X_{new}' = X + F \times (X_b - X) \qquad (8)$$

In formula (8), X_{new}' is the new solution, X is the current solution, is the optimal solution and F is the dynamic factor.

4.2 Quantum Algorithm

It is expressed through quantum's state vector, quantum's bit encoding is used to show the individual of ant colony, and quantum rotation gate and quantum non-gate are used to realize the renewal of individual in ant colony, thus optimization of the target problem can be achieved.

(1) Quantum Bit Representation

In quantum ant colony optimization, the individual ant colony is not represented with a certain value but described through quantum. The range shown with quantum bits can be of any intermediate state among the two states 0 and 1. N quantum bits can be used to show the state 2^N, and in the quantum ant colony optimization, a quantum is located between $[|1>, |0>]$, so the state of a quantum bit can be shown as follows:

$$\begin{aligned} &\phi| = \alpha |0> + \beta |1> \\ &s.t\ \alpha + \beta = 1 \end{aligned} \qquad (9)$$

In formula (9), α represents the probability of $|0>$ and β refers to the probability of $|1>$. In quantum ant colony optimization, the value of a quantum is the smallest

information unit, and the individual ant colony using quantum bit encoding is called individual quantum ant colony. Therefore, the t generation of quantum ant colony is shown as:

$$\kappa(t) = \{\kappa_1^t, \kappa_2^t \cdots \kappa_n^t\} \qquad (10)$$

In formula (10), n refers to the population size, κ_i^t refers to the i quantum individual ant colony in the t generation. Therefore, it can be shown as follows by combing the probability α and β of formula (11):

$$\kappa_i^t = \{\begin{bmatrix} \alpha_{i1}^t \\ \beta_{i1}^t \end{bmatrix}, \begin{bmatrix} \alpha_{i2}^t \\ \beta_{i2}^t \end{bmatrix} \cdots \begin{bmatrix} \alpha_{im}^t \\ \beta_{im}^t \end{bmatrix}\} \qquad (11)$$

By quantum bit encoding, an individual quantum ant colony can represent multiple dynamic superposition at the same time so as to better maintain the diversity of ant colony. And as the value of α and β gets close to 0 and 1 gradually, diversity of colony is also reduced with algorithm convergence.

(2) Quantum Rotation Gate

In quantum ant colony optimization, because there are states in which the individual ant colonies are super-positioned or intertwined, quantum rotation gate is used to process the superposition or intertwining of individual ant colony separately. The generation of descendant individuals mostly comes from and is determined by the optimal individual of the parent generation and probability of states. Therefore, quantum rotation gate mainly functions on the superposition or intertwining state of quantum, which changes in the process of mutual interference so as to change their respective states. Quantum rotation gate is as shown in the follows:

$$\omega = \begin{bmatrix} \sin a & \cos a \\ \sin a & \tan a \end{bmatrix} \qquad (12)$$

4.3. Algorithm Steps

When the quantum self-adaptive ant colony is used to generate the testing cases, an appropriate mapping way must be found firstly. The process of generating path testing data of the program may be subject to the influence of testing parameters, and these factors mainly include entrance parameter of the testing module, global variable of the testing module and internal variables of the testing module, etc. In order to better test the multipath coverage testing case, there is impossible to be only one parameter to be set. When there are multiple parameters input, each of the parameter input should be encoded into a binary separately, and then the parameter is connected to form a separate entity. And then it is encoded by self-adaptive algorithm and quantum bit algorithm.

(1) Initialize the individual: find the variables to be generated into testing data through given global path, encode it according to formula (1), and then combine them to form an individual.

(2) Calculate the self-adaptive value of each individual ant colony according to self-adaptive function. If it meets the condition, go to step (6), and if not, implement step (3)

(3) Improve the individual ant colony according to quantum algorithm, and choose the individual of the nest generation according to the self-adaptive value.

(4) Superposition of solutions: make single superposition of two individuals from the generated individuals so as to obtain new individual until all the solutions have been super-positioned.

(5) Intertwining of solutions: process the intertwining emerges aiming at the superposition process of solution through ω in quantum gate.

(6) Transfer the corresponding data of ant colony meeting the condition in split into testing data, which will become new testing data, and the data testing is completed and finished.

5. Simulation Experiment

Hardware in this paper is Core i3 with the hardware environment of main frequency as 2.4G, memory as 4GDDR3, hardware as 500G and operation system as Window 8 as the testing verification. Matlab2010 is used as the simulation platform, and multipath testing case diagram in control flow diagram is used as the basis of experiments in testing path. It is as shown in Figure 1:

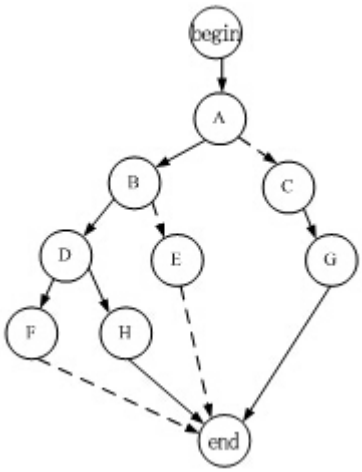

Figure 1. Control Flow Diagram

Solid lines in the figure refer to the control flow with true logic, and dotted lines refer to the control flow with false logic. Herein, logic numbers in the figure are shown in the following Table 1:

Table 1. Testing Path

Path	Route
1	ABDF
2	ABDH
3	ABE
4	ACG

The number of colony is set as 100 with threshold value as 0.3, compare the four paths in table 1 of the same iteration times with algorithm in this paper and basic ant colony

optimization respectively, and the comparison is shown in Figure 2-5:

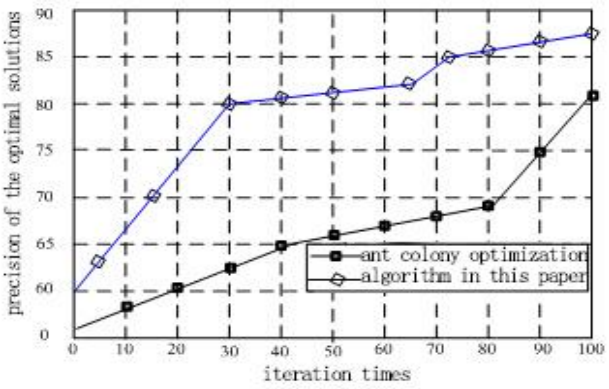

Figure 2. Precision of Path1's Optimal Solution when the Iteration Times Is100

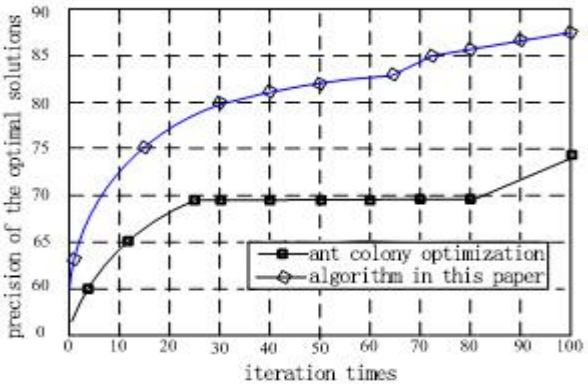

Figure 3. Precision of Path2's Optimal Solution when the Iteration Times Is100

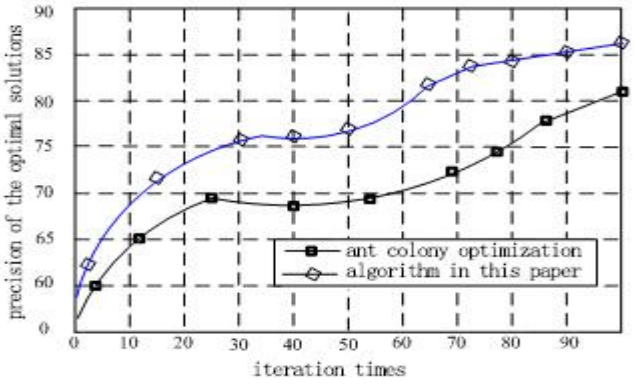

Figure 4. Precision of Path3's Optimal Solution when the Iteration Times Is100

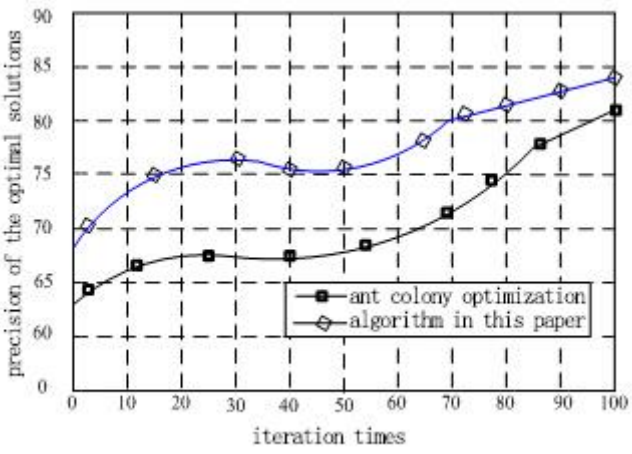

Figure 5. Precision of Path4's Optimal Solution When the Iteration Times Is100

It can be found from Figure 2-5 that function of self-adaptive factors is used in the algorithm in this paper for optimization, and quantum algorithm is combined during the optimization so as to improve the precision of the optimal solutions to the automatic data in the algorithm in this paper, better avoid the influence of the distribution characteristics of program's input variables and greatly improve the effectiveness of the testing.

6. Conclusion

During the automatic generation of testing data, although ant colony optimization has some certain effects, there exist some defects and limitations in the algorithm itself. Based on this, the introduction of self-adaptive factors and quantum algorithm has improve the defects and shortcomings of ant colony optimization, so they have strong use value to the automatic generation of testing data in software testing. Bedsides, simulation experiments show that the testing case generation efficiency generated by the algorithm in this paper is higher than ant colony optimization, so it has certain promotion value.

References

[1] Fang Binxing,Lu Tianbo,LiChao.Survey of softwave assurance[J].Journal on Communications,2009,30(2):106-117

[2] Hermadi I, Ahmed M A. Genetic algorithm bsed test data generator[C] //Evolutionary Computation, 2003.

[3] Korel B. Automated software test data generation.Software Engineering [J].IEEE Transactions, 1990,16(8):870-879.

[4] Fu Bo.Automated Software Test Data Generation Based on Simulated Annealing Genetic Algorithms[J]. Computer Engineering and Applications，2011（12）：82-84.

[5] LiJun，Li yan-hui，Peng yin-cun. Path Test Data Generation Based on Adaptive Genetic Algorithm [J]. Computer Engineering，2010, 35（2）：203-205.

[6] XiaYun，Liu Feng.Automated software test data generation based on immune genetic algorithm [J].Computer Applications，2012（3）：723-725.

[7] Li ai-guo，Zhang yan-li. Automatic Generation Method of Test Data for Software Structure Based on PSO [J]. Computer Engineering，2013（6）：93-94, 97.

[8] GUTJAHRWJ.Agraph-basedantsystemanditseonversenee[J].FutureGenerationComPuterSystem,2013(X),16(8):873-888.

Authors

Chen Ping (1978.7-), male, Huaining, Anhui, master, Associate Professor, main research direction: software engineering.

Xia Min (1977.11-), female, Ma Anshan, Anhui, Master, Lecturer, main research direction: software engineering.

