Evolution test by improved genetic algorithm with application to performance limit evaluation of automatic parallel parking system

Feng Gao1,2 | Qiang Zhang1 | Zaidao Han1 | Yiheng Yang3

1 School of Automotive Engineering, Chongqing University, Chongqing, China
2 Shanghai Jiao Tong University Sichuan Research Institute, Chengdu, China
3 School of Electrical Engineering, Chongqing University, Chongqing, China

Correspondence
Feng Gao.
Email: gaofeng1@cqu.edu.cn

Funding information
Natural Science Foundation of Chongqing, Grant/Award Number: cstc2019jcsj-zdxmX0018; Sichuan Science and Technology Program, Grant/Award Number: 2020YSY070

Abstract
Performance limit evaluation of automatic driving system before putting into the market is critical for driving safety. The evolution test by genetic algorithm (GA) is a method by iteratively generating new test scenarios according to the last test results. To avoid its blind search for better efficiency, a scenario complexity index is proposed to measure the test effectiveness indirectly and guide the evolution process under the assumption that a complex scenario is more challenging to realise automatic driving. The traditional crossover and mutation operators are modified to generate more complex scenarios to improve the test efficiency. The advantage of the improved crossover/mutation operators in increasing the offspring’s scenario complexity index is analysed in theory. Moreover, the influence of the design parameters on the evolution test process and the global convergence are also discussed. The new evolution test by this improved GA has been applied to find the collision condition of a parallel automatic parking system to validate its effectiveness.

1 INTRODUCTION

Compared with traditional onboard systems, automatic driving brings much better social and economic benefits, so it becomes a hot spot for research and production [1–3]. Many competitions have been held to promote its commercialisation, for example, the Grand Challenge sponsored by the Defense Advanced Research Projects Agency [4] and the Intelligent Vehicle Future Challenges sponsored by the National Science Foundation of China [5]. Such competitions catch the public’s attention and try to show the maturity of automatic driving technologies, but only typical use scenarios are considered. It is far away to ensure the reliability and performance of the automatic driving system (ADS), which relates to traffic safety directly and are influenced by uncontrolled traffic environments [6, 7].

Field test [8] and road test [9] are widely used methods to validate the developed system, but they are time-consuming and costly. With the help of X-in-the-loop technologies [10, 11], some researchers tested ADS by replaying the recorded real traffic data to save time and money [6, 12]. Compared with vehicle test, it also has the advantage of safety and repeatability. However, its effectiveness is limited by the completeness of the data, and the tested ADS works in open loop essentially [13]. Besides, there exist many similar and simple scenarios in natural driving conditions, which are inefficient to find the hidden problems.

To realise an effective closed-loop test, some systematic methods have been adopted to design test scenarios [14]. The full coverage of the selected scenario elements can be ensured by the combinational theory [15]. Moreover, if an analytic expression of the control logic can be established, the test can be accelerated by resampling the scenario elements to increase the probability of critical conditions [16]. However, most ADS algorithms are too complicated to be described by an analytic expression, and it is also impractical for suppliers to open their confidential logic to manufacture.

To relax the dependence on the prior information of ADS, an accelerated test method by interactively optimising the probability of scenario elements using the cross-entropy method was proposed in [17, 18]. This approach can find the optimal scenario that is most likely to detect the performance limit. It focuses on the probability distribution instead of the specific value of the scenario element. This is regarded as an indirect optimisation method because after finding the optimal distribution, we still need to look for the exact scenario, which activates the performance limit. So some researchers proposed an idea of direct optimisation, which considers the scenario
element as optimisation parameters and iteratively updates their values to generate new test scenarios by intelligent algorithms [5, 19]. Klück et al. compared the efficiency of two frequently used intelligent algorithms, that is, genetic algorithm (GA) and simulated annealing, in performance limit evaluation of ADS through a number of experiments and concluded that GA is more efficient [20]. Vos et al. carried out an evolution test by GA for an adaptive cruise control system [21]. Furthermore, when doing the evolution test by GA for an autonomous parking system, Bühler et al. introduced a novel fitness function to accelerate the optimisation process [22]. These studies are beneficial to test efficiency, but the evolution process is still random because the effectiveness of the generated offspring cannot be estimated beforehand.

To reduce the blindness of searching, an index is designed to measure the scenario complexity under the assumption that automatic driving is more difficult to be realised under complex scenarios. The offspring generation process of the crossover and mutation operators are improved by the guidance of this scenario complexity index. With the idea of brother competition, the candidate offspring generated by the new crossover operator competes to survive according to the average scenario complexity index, and so will the candidate offspring generated by the new mutation operator. In this way, the generated offspring becomes more complex, which is beneficial to test efficiency. After the improvement analysis of the scenario complexity theoretically, the effectiveness of the evolution test by this improved GA is validated by application on an automatic parallel parking system (APPS) to evaluate its collision avoidance performance.

The main contributions of this paper are:

1. A scenario complexity index is proposed to measure the effectiveness of the test scenario without conducting the test;
2. the traditional crossover and mutation operators of GA have been re-designed to increase the searching speed by introducing the scenario complexity index;
3. the improvement of the offspring’s scenario complexity index and the influence of the key design parameters have been analysed theoretically;
4. The effectiveness of the evolution test by the improved GA has been validated comparatively by application on APPS.

The remained parts are organised as follow. The main process is introduced in Section 2. Sections 3 and 4 design the new crossover and mutation operators, respectively. In Section 5, the performance is studied from: (1) The influence of the key parameter on the evolution test and (2) the global convergence. The effectiveness is validated in Section 6, and Section 7 concludes the paper.

2 | EVOLUTION TEST BY IMPROVED GA

Just like the traditional test, the evolution test by the improved GA is also composed of five steps as shown in Figure 1 [21], where the traditional crossover/mutation operators are modified to improve test efficiency. The new ones are called full crossover and multiple mutation, respectively. With the assumption that the performance limit is more likely to be activated with complex scenarios, the full crossover and multiple mutation operators are designed to increase the possibility of producing more complex scenarios. The details are introduced in Sections 3 and 4, respectively.

The initial population, \( X_1 = \{X_{1,1}, \ldots, X_{1,n}\} \) with 2\( n \) test scenarios, is generated randomly. The test scenario, \( X_m \), is composed of the values selected randomly from the \( m \) scenario elements. Each scenario is considered as an individual and denoted by a chromosome with \( L \) genes. The performance is measured by an objective function. When it reaches the desired value, it implies that the performance limit is found and the test process is terminated. For example, if the test target is to detect a collision, the distance from the obstacle can act as the objective function. Moreover, due to the limit of test time, when the number of iterations is larger than the allowable one, the test is also terminated in practice.

\[
\min_{X \in R} h(X),
\]

where \( X \) is the optimisation variables, that is, the test scenario; \( h(X) \) is the objective function measuring the desired performance; \( R \) is the set composed of all possible test scenarios. In Expression (1), the performance limit is assumed to be the minimum value of the objective function as an example. For the application whose performance limit is the maximum, the optimisation problem can be converted to Expression (1) equivalently. The objective function is designed according to the desired performance of the tested ADS, which is used to evaluate its behaviour and is always provided by the supplier.
The crossover and mutation operators are cores of GA. To improve them by increasing the possibility of producing more complex scenarios for better efficiency, an index is required to measure the effectiveness of the test scenario without conducting the test. But it is hard to set up an analytical function between the performance of ADS and the test scenario. To overcome this challenge, the analytic hierarchy process (AHP) is adopted to design the scenario complexity index [14, 15]. Being different from the objective function in Expression (1), the scenario complexity index is an estimation of the effectiveness of the test scenario, that is, the ability of the test scenario to activate the performance limit of the tested ADS.

As shown in Figure 2, to realise the evaluation of the test effect indirectly, a tree model is built first to analyse the influence of scenario elements, which affect the functionality and performance of the tested ADS. In Figure 2, $b_{i,j}$ is the $j$-th node in the $i$-th layer. In layer 1 to layer $F-1$, each node represents a scenario element, while in the bottom layer, each node is one value of its parent node. Since the number of nodes is limited, some discretisation methods, such as equivalence partitioning and boundary value analysis, can be adopted to convert the continuous element to limited and discrete ones [26, 27].

The influence factors of ADS are numerous and complicated. It is very difficult to determine the importance of each factor when designing the test scenario quantitatively. By this tree structure model, only comparative analysis among the factors belonging to the same parent node is needed. Then, the importance degree of the node at the bottom layer is obtained by AHP [28, 29]:

$$T_k = \prod_{(i,j) \in R} R_{i,j}$$

where $T_k$ is the importance degree, $R_{i,j}$ is the relative importance of node, $b_{i,j}$, and $R$ is a set composed of the indexes of the path from $b_{F,k}$ to the root.

A test scenario, $X$, is represented by the values of scenario elements, that is, $X = \{x_1, ..., x_m\}$, where $x_i$ represents the value of the $i$-th scenario element. To find the exact performance limit, $x_i$ is generated randomly in its continuous range, so $x_i$ is not equal to the discrete value, $b_{F,i}$. The linear interpolation is used to calculate the important degree of $x_i \in \{b_{F,k-1}, b_{F,k}\}$ and the scenario complexity index is obtained by summarisation:

$$C(X) = \sum_{i=1}^{m} T_i$$

$$T_i = T_{k-1} \times \frac{x_i - b_{F,k-1}}{b_{F,k} - b_{F,k-1}} \times (T_k - T_{k-1})$$

where $C(X)$ is the scenario complexity index.
FIGURE 4  Multiple mutation operator

according to their scenario complexity index, which has a positive correlation with the selection probability.

As shown in Figure 3, the pseudocode to realise the full crossover operator is

1. \( x_i^C (1 \leq j \leq L) \) is selected as offspring pair and then 
\[
\mathbb{P} (x_i^C = x_{i,j}^C) = \omega / L, \text{ where } \omega \text{ is the crossover probability};
\]
2. \( y_i^P \) is selected as offspring pair directly and then 
\[
\mathbb{P} (\hat{x}_i^C = \hat{y}_i^P) = 1 - \omega.
\]

According to the above two conditions, the following theorem shows the relationship of the expect of offspring’s scenario complexity index between the traditional and modified crossover operators.

**Theorem 1:** Compared with the single-point crossover operator with the crossover probability \( \omega \), the expect of offspring’s scenario complexity index generated by the full crossover operator satisfies:

- (C1) If \( \omega = 1 \), \( \mathbb{E}(\hat{x}_i^C) \geq \mathbb{E}(\hat{y}_i^C) \).
- (C2) If \( \omega \in [0, 1] \) and the average scenario complexity index of all candidate offspring pairs is not smaller than that of their parent, that is, \( \frac{1}{L} \sum_j = 1 \mathbb{C}(x_i^C) \geq \mathbb{C}(y_i^P) \), then \( \mathbb{E}(\mathbb{C}(x_i^C)) \geq \mathbb{E}(\mathbb{C}(y_i^P)) \).

**Proof:** The proof is in Appendix A.2.

Theorem 1 shows the improvement of the offspring’s scenario complexity index by the full crossover operator. This is beneficial to find out the critical condition. But it is known from C2 that when \( \omega \) is small enough, the full crossover operator may generate offspring with a smaller scenario complexity index. However, this situation appears hardly in practice because \( \omega \) cannot be too small, otherwise it easily causes premature [24].

### 4  |  MULTIPLE MUTATION OPERATOR

#### 4.1  |  Design of multiple mutation operator

The traditional operator only generates one population, among which it has a low possibility to include the good individuals. To increase this possibility, the multiple mutation operator is designed as shown in Figure 4. It performs traditional canonical mutation [30] on the population for \( N \) times. To keep the population size, the brother competition strategy is adopted to select the formal offspring according to their overall scenario complexity index.

According to the multiple mutation operator shown in Figure 4, its pseudocode is

1. \( \text{For } i = 1 \text{ to } N \)
2. \( x_i^M = \text{Mut}(x_i^C, w_M) \)
3. End
4. Randomly select a candidate mutation offspring population as the formal one \( X_M^C \) with the probability that
\[
x_i^M = e^{\alpha \mathbb{C}(x_i^C)} / \sum_{k=1}^L e^{\alpha \mathbb{C}(x_k^C)}
\]

where \( \mathbb{C}(x_i^C, w_M) \) denotes the canonical mutation on \( X_C^C \) with the mutation probability, \( w_M \).
4.2 Analysis of offspring’s scenario complexity index

It is hard to compare the expect of scenario complexity index directly. To analyse the improvement of the scenario complexity index, a new operator named blind multiple mutation is introduced to act as a bridge between the traditional operator and the multiple mutation operator. Its definition and characteristic are introduced as the following.

Blind multiple mutation operator (BMMO): It performs the canonical mutation operation on a population for $N$ times, and then selects one of them as the formal one with an equal probability of $1/N$.

**Lemma 1:** From a statistical point of view, BMMO is equal to the canonical mutation operator.

**Proof:** The proof is in Appendix A.3.

With help of BMMO, the following theorem is established to show the improvement of scenario complexity index by the multiple mutation operator.

**Theorem 2:** Compared with the canonical mutation, the expect of scenario complexity index of the offspring generated by the multiple mutation operator satisfies:

$$E\left(\bar{C}(X^M)\right) \geq E\left(\bar{C}(X^M)\right) \quad (6)$$

**Proof:** The proof is in Appendix A.4.

Theorem 2 shows that the multiple mutation operator ensures that the expect of the offspring’s scenario complexity index is not smaller than the canonical one. It is concluded from Theorems 1 and 2 that the improved GA has a higher possibility to generate the offspring with a higher scenario complexity index after each iteration. Since the driving task is more difficult to be executed in complex scenarios, the evolution test by the improved GA is easier to find the critical condition.

5 PERFORMANCE ANALYSIS OF IMPROVED GA

5.1 Influence analysis of design parameters

From the pseudocode of full crossover operator (Line (6)) and Equation (5), the following equation establishes:

$$E\left(\bar{C}(X^c)\right) = \frac{\sum_{j=1}^{L_c} e^{\alpha xC(X^c_{i,j})} \times \bar{C}(X^c_{i,j})}{\sum_{k=1}^{L_c} e^{\alpha xC(X^c_{i,k})}} \quad (7)$$

Similarly, from the pseudocode of the multiple mutation operator (Line 4) and Equation (21), we have

$$E\left(\bar{C}(X^M)\right) = \frac{\sum_{i=1}^{N} e^{\alpha xC(X^M_{i,j})} \times \bar{C}(X^M_{i,j})}{\sum_{k=1}^{N} e^{\alpha xC(X^M_{i,k})}} \quad (8)$$

It is known from Equations (7) and (8) that the design parameter, $\alpha$, influences $E(\bar{C}(X^c_{i,j}))$ and $E(\bar{C}(X^M_{i,j}))$ directly. To study their relationship for the selection of $\alpha$, a function, $g(\alpha)$, is introduced:

$$g(\alpha) = \sum_{i=1}^{L_c} e^{\alpha x} / \sum_{i=1}^{N} e^{\alpha x} \quad (9)$$

It is found by comparing Equations (9) with (7) and (8) that the monotonic relation between $\alpha$ and $g(\alpha)$ is the same as that between $\alpha$ and $E(\bar{C}(X^c_{i,j}))$ or $E(\bar{C}(X^M_{i,j}))$. The following lemma is introduced first by using Equation (9) for the influence analysis of $\alpha$.

**Lemma 2:** The relation between $\alpha$ and $g(\alpha)$ satisfies:

1. $\alpha \rightarrow \infty$, $g(\alpha)$ is not always beneficial to test, but has an upper bound. However, in practice, a larger $\alpha$ is not always beneficial to test efficiency because the population convergences to the area with the highest local scenario complexity index more easily if $\alpha$ is too large.

5.2 Convergence of improved GA

The following definition and lemma about the convergence of GA are introduced first.

**Definition [30]:** If all elements in matrix $B = [b_{ij}]_{N \times 1}$ are constant and $B$ satisfies:

1. It has at least one positive entry in each column, then it is called a column-allowable matrix;
2. for all $i \in \{1, 2, ..., l\}$, $\sum_{j=1}^{l} b_{ij} = 1$ establishes, then it is called a stochastic matrix;
3. for all $i, j \in \{1, 2, ..., l\}$, $b_{ij} > 0$ establishes, then it is called a positive matrix.

**Lemma 3 [30]:** For the canonical GA with the assumption that it is a Markov chain, the global convergence is achieved by the elitist selection strategy, if the transition matrix for selection is column-allowable, that of the crossover operator is stochastic and that of the mutation operator is positive.

Based on Lemma 3, the following theorem ensures the global convergence of the improved GA.

**Theorem 4:** The improved GA converges to the global optimum if the process is a Markov chain.

**Proof:** The proof is in Appendix A.7.
6 APPLICATION AND ANALYSIS

The proposed evolution test by the improved GA can be used for different intelligent driving systems by properly defining the objective function, $h(X)$, and parameterising the test scenario to measure its complexity. Here, it is applied to the performance limit evaluation of an APPS in avoiding collision with the surroundings to show its effectiveness as an example.

6.1 Experiment platform and scenario

The experiment was conducted by a model-in-loop test platform shown in Figure 5 [31]. The virtual scenarios and sensors are modelled in PreScan. The parallel parking control algorithm and vehicle dynamics model were run in Matlab/Simulink. The improved GA is programed by m-script of Matlab. It calls PreScan and Matlab/Simulink to generate the closed-loop simulation system and conduct the simulation automatically.

The schematic of the test scenario is shown in Figure 6. In this study, we focus on the evolution test algorithm, and there is only one vehicle acting as the surrounding object. The test objective is to find the minimum distance between the tested vehicle and the surroundings. Of course, there exist several types of performances and functionalities of APPS. In this study, only the collision avoidance performance is considered as an example to validate the effectiveness of the evolution test by the improved GA. If more performances or functionalities need to be tested, the objective function in Expression (1) should be re-designed to describe them.

According to the tree structure model introduced in Section 2.2, the considered scenario elements are listed in Table 1. All scenario elements are grouped into three types according to their physical meaning, that is, ‘orientation’, ‘distance’ and ‘velocity’. Each scenario element is denoted by $I_i$ and their discrete values are listed in the column labeled by ‘layer 3’. By AHP, the importance degree of each discrete value is obtained as shown in the column labelled by $\lambda_k$ before conducting the test. During the evolution test, if a scenario is generated, its scenario complexity index can be calculated by Equation (4).

6.2 Experiment results and analysis

To validate the hypothesis that the test scenario with a higher complexity index is more likely to detect the performance limit, the test results and the scenario complexity index of the generated 413 scenarios are analysed statistically as shown in Figure 7. From a statistical point of view, the effectiveness of the test scenario has a positive correlation with its scenario complexity index.

With the setting that $n = 5$, $L = 100$, $A = 1.9$, $w_M = 0.009$, $G_{\alpha} = 25$, $\alpha = 600$, $N = 50$ and $w_G = 1$, the overall offspring's scenario complexity index is shown in Figure 8(a). The proposed full crossover and multiple mutation operators are beneficial to increase the offspring’s scenario complexity index. Therefore, the test efficiency is improved (see Figure 8(b)). Both the convergence and the ability to find the
collision condition are better than traditional GA. The improved GA detects the collision situation at the 11th generation. On the contrary, the worst condition found by traditional GA is that the minimum distance is about 0.1 m at the 25th generation.

### 6.3 Influence analysis of parameters

The two important parameters of the improved GA are $\alpha$ and $N$ (see the pseudocode in Sections 3 and 4). Intuitively, $\alpha$ has a positive correlation with the tendency of selecting the candidate offspring with a higher scenario complexity index, and a larger $N$ will generate more candidate offspring population.

To show their influence, the evolution test has been conducted with different parameter values. The experiment was conducted six times for each value to avoid randomness. The average test results are shown in Tables 2 and 3.

It is found from Table 2 that with the increase of $\alpha$, the test effect becomes better until $\alpha = 400$. After that, the result tends to be stable. This is caused by the saturation of the influence of scenario complexity on evolution. According to the results in Table 3, when $N$ is small, the test effect is not good enough because the probability that the good offspring appears in the candidate offspring population is low. As $N$ increases, this probability becomes higher, and so the test effect also becomes better. When $N$ reaches 30, the test effect stays stably. The reason is that when the number of samples is large enough, the distribution of scenario complexity index of the candidate offspring population tends to be stable.

### 7 Conclusion

In this paper, an evolution test by the improved GA is presented for the performance limit evaluation of ADS to increase the test efficiency. Both theoretical analysis and application results show that:

1. The proposed scenario complexity index can measure the test effectiveness without conducting the test statistically;
2. the scenario complexity index is beneficial to guide the generation of offspring, and the proposed full crossover and multiple mutation operators can generate more complex scenarios;
3. the ability to find the performance limit is improved by replacing the traditional operators with the full crossover and multiple mutation operators.

| $\alpha$ | 100 | 200 | 400 | 600 | 800 |
|---------|-----|-----|-----|-----|-----|
| Convergence distance (m) | 0.09 | 0.03 | 0.02 | 0.02 |
| Convergence generation | 20.17 | 16.00 | 9.50 | 9.00 |

| $N$ | 5 | 10 | 30 | 40 | 50 |
|-----|---|----|----|----|----|
| Convergence distance (m) | 0.14 | 0.05 | 0.04 | 0.02 |
| Convergence generation | 18.33 | 13.67 | 8.17 | 8.67 | 9.00 |
ACKNOWLEDGEMENTS

This work was supported by the Natural Science Foundation of Chongqing under grant cstc2019jcyj-zdxmX0018 and the Sichuan Science and Technology Program under grant 2020YFSY0070.

REFERENCES

1. Li, K., et al.: Robust distributed consensus control of uncertain multi-agents interacted by eigenvalue-bounded topologies. IEEE Internet Things J. 7(5), 3790–3798 (2020)
2. Tang, F., Gao, F., Wang, Z.: Driving capability based transition strategy for cooperative driving: From manual to automatic. IEEE Access 8, 139013–139022 (2020)
3. Gao, F., Dang, D., He, Y.: Robust coordinated control of nonlinear heterogeneous platoon interacted by uncertain topology. IEEE Trans. Intell. Transp. Syst. 11(2) (2020). https://doi.org/10.1109/TITS.2020.3045107
4. Campbell, M., et al.: Autonomous driving in urban environments: Approaches, lessons and challenges. Philos. Trans. R. Soc. A 368(1928), 4649-4672 (2010)
5. Li, L., et al.: Artificial intelligence test: A case study of intelligent vehicles. Artif. Intell. Rev. 50(3), 441–465 (2018)
6. Dang, D., Gao, F., Hu, Q.: Motion planning for autonomous vehicles considering longitudinal and lateral dynamics coupling. Appl. Sci. 10(9), 3180 (2020)
7. Gao, F., Wang, C.: Hybrid strategy for traffic light detection by combining classical and self-learning detectors. IET Intel. Transport Syst. 14(7), 735–741 (2020)
8. Metcy Test Facility. https://metcy.umiich.edu/our-work/metcy-test-facility/edu/test-facility (2019). Accessed 11 June 2019
9. Self-Driving Cars & Bikes: An Interview With Google. https://bikeaustin.org/2015/09/self-driving-cars-bikes-an-interview-with-google. Accessed 11 June 2019
10. Duan, J., Gao, F., He, Y.: Test scenario generation and optimization technology for intelligent driving systems. IEEE Trans. Intell. Syst. Mag. 1-1 (2020). https://doi.org/10.1109/MITSS.2019.2926269
11. Xia, Q., et al.: Decoupled H-infty control of automated vehicular platoons with complex interaction topologies. IET Intel. Transport Syst. 11(2), 92–107 (2017)
12. Li, X., et al.: ParallelEye-CS: A new dataset of synthetic images for testing the visual intelligence of intelligent vehicles. IEEE Trans. Veh. Technol. 68(10), 9619-9631 (2019)
13. Kang, Y., Yin, H., Berger, C.: Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments. IEEE Trans. Intell. Veh. 4(2), 171–185 (2019)
14. Xia, Q., et al.: Test scenario design for intelligent driving system ensuring coverage and effectiveness. Int. J. Automot. Technol. 19(4), 751–758 (2018)
15. Gao, F., et al.: Test scenario automatic generation strategy for intelligent driving systems. Math. Probl. Eng. 2019, 759450 (2019)
16. Zhao, D., et al.: Accelerated evaluation of automated vehicles in car-following maneuvers. IEEE Trans. Intell. Syst. 19(3), 733–744 (2018)
17. Zhao, D., et al.: Accelerated evaluation of automated vehicles safety in lane change scenarios based on importance sampling techniques. IEEE Trans. Intell. Transp. Syst. 18(3), 595–607 (2017)
18. Huang, Z., et al.: Accelerated evaluation of automated vehicles using piece-wise mixture models. IEEE Trans. Intell. Transp. Syst. 19(9), 2845–2855 (2018)
19. Vishnukumar, H.J., et al.: Machine learning and deep neural network—Artificial intelligence core for lab and real-world test and validation for ADAS and autonomous vehicles: AI for efficient and quality test and validation. In: Proceedings of Intelligent Systems Conference, London, UK, pp. 714–721 (2017)
20. Kluck, F., et al.: Performance comparison of two search-based testing strategies for ADAS system validation. In: Proceedings of 31st International Conference on Testing Software and Systems, Paris, France, pp. 140–156 (2019)
21. Vos, T.E.J., et al.: Evolutionary functional black-box testing in an industrial setting. Software Qual. J. 21(2), 259–288 (2013)
22. Bühler, O., Wegener, J.: Automatic testing of an autonomous parking system using evolutionary computation. SAE Technical Paper no. 2004-01-0459 (2004)
23. Bäck, T., Hoffmeister, F.: Extended selection mechanisms in genetic algorithms. In: Proceedings of 4th International Conference on Genetic Algorithms, San Diego, USA, pp. 92–99 (1991)
24. Li, J., Li, M.: An analysis on convergence and convergence rate estimate of genetic algorithms in noisy environments. Optik 124(24), 6780–6785 (2013)
25. Peng, Y., Luo, X., Wei, W.: A new fuzzy adaptive simulated annealing genetic algorithm and its convergence analysis and convergence rate estimation. Int. J. Control Autom. Syst. 12(3), 670–679 (2014)
26. De Ona R., De Ona, J.: Analysis of transit quality of service through segmentation and classification tree techniques. Transportmetrica A: Transport Sci. 11(5), 365–387 (2015)
27. Barhate, S.S.: Effective test strategy for testing automotive software. In: Proceedings of International Conference on Industrial Instrumentation and Control, Pune, India, pp. 645–649 (2015)
28. Saaty, T.L.: Axiomatic foundation of the analytic hierarchy process. Manage. Sci. 32(7), 841–855 (1986)
29. Ishizaka, A., Labib, A.: Review of the main developments in the analytic hierarchy process. Expert Syst. Appl. 38(11), 14336–14345 (2011)
30. Rudolph, G.: Convergence analysis of canonical genetic algorithms. IEEE Trans. Neural Networks 5(1), 96–101 (1994)
31. Gao, F., et al.: Automatic virtual test technology for intelligent driving systems considering both coverage and efficiency. IEEE Trans. Veh. Technol. 69(12), 14365–14376 (2020)

How to cite this article: Gao F, Zhang Q, Han Z, Yang Y. Evolution test by improved genetic algorithm with application to performance limit evaluation of automatic parallel parking system. IET Intell Transp Syst. 2021;15:754–764. https://doi.org/10.1049/itr2.12058.

APPENDIX A

A.1 | Definitions of symbols

\( X_i^P \): The \( i \)-th parental individual pair

\( X_i^j \): The candidate offspring pair generated by conducting the single-point crossover on \( X_i^P \) at the \( j \)-th position

\( w_i^C \): The probability of selecting \( X_i^C \) as the formal crossover offspring

\( X_i^C \): The formal offspring of \( X_i^P \) generated by the full crossover

\( X_i^C \): The offspring generated by conducting the traditional single-point crossover on \( X_i^P \)

\( X_i^C = \{ X_i^1, \ldots, X_i^c \} \): The formal offspring population generated by the full crossover

\( X_i^A \): The candidate offspring population generated by conducting the \( i \)-th canonical mutation

\( w_i^A \): The probability of selecting \( X_i^A \) as the formal mutation offspring

\( X_i^A \): The formal offspring population generated by the multiple mutation

\( X_i^A \): The mutation offspring population generated by the canonical mutation
A.2 Proof of Theorem 1

According to the algorithm of single-point crossover, its scenario complexity index of the offspring is

\[ E\left( C\left( \hat{X}_i^C \right) \right) = \frac{\omega_c L \sum_{j=1}^{L} C\left( \hat{X}_{i,j}^C \right)}{L} + (1 - \omega_c) C\left( \hat{X}_i^P \right) \quad (10) \]

Subtracting Equations (10) from (5) yields

\[ E\left( C\left( \hat{X}_i^C \right) \right) - E\left( C\left( \hat{X}_i^C \right) \right) = \sum_{j=1}^{L} \left[ \left( w_{i,j}^C - \frac{1}{L} \right) C\left( \hat{X}_{i,j}^C \right) \right] - (1 - \omega_c) C\left( \hat{X}_i^P \right) \quad (11) \]

When \( \omega_c = 1 \), Equation (11) is rewritten as

\[ E\left( C\left( \hat{X}_i^C \right) \right) - E\left( C\left( \hat{X}_i^C \right) \right) = \sum_{j=1}^{L} \left( w_{i,j}^C - \frac{1}{L} \right) C\left( \hat{X}_{i,j}^C \right) \quad (12) \]

To facilitate the analysis, according to whether the full crossover operator is able to improve the probability to generate \( X_{i,j}^C \), the right side of Equations (12) is divided into two parts:

\[ E\left( C\left( \hat{X}_i^C \right) \right) - E\left( C\left( \hat{X}_i^C \right) \right) = \sum_{j \in S_+^C} \left( w_{i,j}^C - \frac{1}{L} \right) C\left( \hat{X}_{i,j}^C \right) \]

\[ + \sum_{j \in S_-^C} \left( w_{i,j}^C - \frac{1}{L} \right) C\left( \hat{X}_{i,j}^C \right) \geq \min_{j \in S_+^C} \left( C\left( \hat{X}_{i,j}^C \right) \right) \sum_{j \in S_+^C} \left( w_{i,j}^C - \frac{1}{L} \right) \]

\[ + \max_{j \in S_-^C} \left( C\left( \hat{X}_{i,j}^C \right) \right) \sum_{j \in S_-^C} \left( w_{i,j}^C - \frac{1}{L} \right) \quad (13) \]

where \( S_+^C = \{ j | w_{i,j}^C \geq \frac{1}{L} \} \) and \( S_-^C = \{ j | w_{i,j}^C < \frac{1}{L} \} \).

From the pseudocode (Line 6) of the full crossover, \( \sum_{j=1}^{L} w_{i,j}^C = 1 \), and therefore

\[ \sum_{j \in S_+^C} \left( w_{i,j}^C - \frac{1}{L} \right) + \sum_{j \in S_-^C} \left( w_{i,j}^C - \frac{1}{L} \right) = 0 \]

\[ \Rightarrow \sum_{j \in S_+^C} \left( w_{i,j}^C - \frac{1}{L} \right) = - \sum_{j \in S_-^C} \left( w_{i,j}^C - \frac{1}{L} \right) \quad (14) \]

Then, substituting Equations (14) to (13) yields

\[ E\left( C\left( \hat{X}_i^C \right) \right) - E\left( C\left( \hat{X}_i^C \right) \right) \geq \left[ \min_{j \in S_+^C} \left( C\left( \hat{X}_{i,j}^C \right) \right) - \max_{j \in S_-^C} \left( C\left( \hat{X}_{i,j}^C \right) \right) \right] \]

\[ \times \sum_{j \in S_-^C} \left( w_{i,j}^C - \frac{1}{L} \right) \quad (15) \]

According to the equation of \( u_i^C \) (Line 6 in the pseudocode of the full crossover), it has a positive correlation with \( C(\hat{X}_{i,j}^C) \).

It is concluded from the definition of \( S_+^C \) and \( S_-^C \) that

\[ \min_{j \in S_+^C} \left( C\left( \hat{X}_{i,j}^C \right) \right) \geq - \max_{j \in S_-^C} \left( C\left( \hat{X}_{i,j}^C \right) \right) \geq 0, \quad (16) \]

Then, Theorem 1 (C1) is proved by substituting Equations (16) into (15).

When \( \omega_c \in [0, 1) \), the following equation is obtained by rewriting Equation (14):

\[ 1 - \omega_c = \sum_{j=1}^{L} \left( w_{i,j}^C - \frac{\omega_c}{L} \right) \quad (17) \]

Substituting Equations (17) into (11) yields

\[ E\left( C\left( \hat{X}_i^C \right) \right) - E\left( C\left( \hat{X}_i^C \right) \right) = \sum_{j=1}^{L} \left( C\left( \hat{X}_{i,j}^C \right) \right) - C\left( \hat{X}_i^P \right) \]

\[ - \frac{\omega_c}{L} \sum_{j=1}^{L} \left( C\left( \hat{X}_{i,j}^C \right) \right) - C\left( \hat{X}_i^P \right) \quad (18) \]

According to Equation (18), when \( \frac{1}{L} \sum_{j=1}^{L} C\left( \hat{X}_{i,j}^C \right) \geq C\left( \hat{X}_i^P \right), E(\hat{C}(\hat{X}_i^C)) - E(\hat{C}(\hat{X}_i^P)) \) has a negative correlation with \( \omega_c \). So it is further concluded from C1 that \( E(\hat{C}(\hat{X}_i^C)) \geq E(\hat{C}(\hat{X}_i^P)) \) when \( \omega_c \in [0, 1) \), and Theorem 1 (C2) is proved.

A.3 Proof of Lemma 1

Let \( \mathbb{Q} = \{0, 1\}^L \) denote the individual state space, \( \mathbb{Q} = \{0, 1\}^L \) denote the population state space, \( \mathbb{X}_{M+}^L \) denote a population in \( \mathbb{Q} = \{0, 1\}^L \), \( \mathbb{X}_M^L \) denote the formal offspring population generated by BMMO and \( \mathbb{X}_M^L \) denote the offspring population generated by the canonical mutation operator.
Each operation of BMMO is identical and independent, and so

\[
P(\hat{\mathbf{x}} \mathcal{M} = \mathbf{x}^\mathcal{M}_s) = \sum_{i=0}^{N-1} \left\{ \binom{N}{i} \left[ P(\hat{\mathbf{x}} \mathcal{M} = \mathbf{x}^\mathcal{M}_s) \right]^i \right\} \left[ 1 - P(\hat{\mathbf{x}} \mathcal{M} = \mathbf{x}^\mathcal{M}_s) \right]^{N-i}
\]

\[
(19)
\]

Combining Equations (20) to (19), we have

\[
P(\hat{\mathbf{x}} \mathcal{M} = \mathbf{x}^\mathcal{M}_s) = \sum_{i=0}^{N-1} \left\{ \binom{N}{i} \left[ P(\hat{\mathbf{x}} \mathcal{M} = \mathbf{x}^\mathcal{M}_s) \right]^i \right\} \left[ 1 - P(\hat{\mathbf{x}} \mathcal{M} = \mathbf{x}^\mathcal{M}_s) \right]^{N-i}
\]

where \( \binom{N}{i} = \frac{N!}{i!(N-i)!} \).

The following equation is obtained from Lemma 1:

\[
\sum_{i=1}^{N-1} \left\{ \binom{N-1}{i-1} \left[ P(\hat{\mathbf{x}} \mathcal{M} = \mathbf{x}^\mathcal{M}_s) \right]^{i-1} \left[ 1 - P(\hat{\mathbf{x}} \mathcal{M} = \mathbf{x}^\mathcal{M}_s) \right]^{N-i-1} \right\}
\]

\[
(20)
\]

where \( \mathbb{S}_M^+ = \{ |\alpha| \alpha^\mathcal{M}_i \geq \frac{1}{N} \} \) and \( \mathbb{S}_M^- = \{ |\alpha| \alpha^\mathcal{M}_i < \frac{1}{N} \} \).

From the pseudocode (Line 4) of the multiple mutation operator, it is known that \( \sum_{i=1}^{N-1} \alpha^\mathcal{M}_i = 1 \). Being similar with the process from Equations (14) to (16), we have \( E(\mathbf{x}^\mathcal{M}) - E(\hat{\mathbf{x}} \mathcal{M}) \geq 0 \), and the conclusion is proved.

A.5 Proof of Lemma 2

Proof: The derivative of Equation (9) to \( \alpha \) is

\[
g'_\alpha(\alpha) = g_{\beta} \frac{\alpha}{\sum_{i=1}^{\mathbb{n}} \alpha^\beta_i}
\]

\[
g_{\beta} \alpha = \sum_{i=1}^{\mathbb{n}} \alpha_i \alpha^\beta_i \sum_{i=1}^{\mathbb{n}} \alpha^\beta_i - \sum_{i=1}^{\mathbb{n}} \alpha_i \alpha^\beta_i
\]

The denominator of \( g'_\alpha(\alpha) \) is greater than 0, and so its sign is determined by the numerator, which is rewritten as the following by the mathematical induction:

\[
g_{\beta} \alpha = \sum_{i=1}^{\mathbb{n}} (\alpha_i - \alpha_i)^2 \alpha^\beta_i
\]

\[
(24)
\]

The induction process to obtain Equation (25) is as follows:

Step 1. The following equation establishes when \( s = 2 \):

\[
g_{\beta} \alpha = (\alpha_2 - \alpha_1)^2 \alpha^\beta_2
\]

\[
(25)
\]

Step 2. For \( \forall \mathbb{s} = \mathbb{t} \), we have

\[
g_{\beta} \alpha |s = \mathbb{t} = \sum_{i=1}^{\mathbb{t}} (\alpha_i - \alpha_i)^2 \alpha^\beta_2
\]

\[
(26)
\]

\[
g_{\beta} \alpha = \sum_{i=1}^{\mathbb{t}} (\alpha_i - \alpha_i)^2 \alpha^\beta_2
\]

\[
(27)
\]
Since \( g_d (\alpha | s = t + 1) = g_d (\alpha | s = t + 1) - g_d (\alpha | s = t) + g_d (\alpha | s = t) \), the following equation is obtained by substituting \( s = t \) and \( s = t + 1 \) to \( g_d (\alpha) \), respectively.

\[
g_d (\alpha | s = t + 1) = \sum_{j = 1}^{l} \alpha (x_{i,j} + n_j) + \sum_{j = 1}^{l} n_j e^{\alpha (x_{i,j} + n_j)} - \sum_{j = 1}^{l} 2 n_j e^{\alpha (x_{i,j} + n_j)} + g_d (\alpha | s = t)
\]

Substituting Equations (27) to (28) yields

\[
g_d (\alpha | s = t + 1) = \sum_{j = 1}^{l} \left( n_{j+1} - n_j \right)^2 e^{\alpha (x_{i,j+1} + n_j)} + \sum_{j = 1}^{l} \left( n_{j+1} - n_j \right)^2 e^{\alpha (x_{i,j+1} + n_j)} + g_d (\alpha | s = t)
\]

Equation (28) respectively.

\[
\sum_{i = 1}^{l} p_{i,j} \lim_{n \to +\infty} e^{-\alpha n_{\max_{1 \leq i \leq l}}} = \max_{1 \leq i \leq l} n_i
\]

A.6 | Proof of Theorem 3

According to Lemma 2, Theorem 3 can be proved by replacing \( n_i \) in Equation (9) with \( C_i (X_{i,j}^C) \) in Equation (7) and \( C_i (X_{i,j}^M) \) in Equation (8) respectively.

A.7 | Proof of Theorem 4

The elitist selection strategy is used by the improved GA. It is known from Lemma 3 that the global convergence can be ensured by proving that the transition matrices of the natural selection, the full crossover operator and the multiple mutation are column-allowable, stochastic and positive, respectively.

Let \( S = \{p_{i,j}^S\}_{Q_{2^i} \times \{Q_{2^i} \}} \) denote the transition matrix of the natural selection, where \( p_{i,j}^S \) is the probability that the population is transferred from state \( i \) to \( j \). Then, \( S \) is column-allowable because

\[
p_{i,j}^S \geq \prod_{k = 1}^{2^n} \left( A - 2 (A - 1) (k - 1) / (2^n - 1) \right) > 0
\]

Let \( C = \{p_{i,j}^C\}_{Q_{2^i} \times \{Q_{2^i} \}} \) denote the transition matrix of the full crossover operator, where \( p_{i,j}^C \) denotes the probability that the population is transferred from state \( i \) to \( j \). The value of \( p_{i,j}^C \) only depends on the gene sequence of the parental population and the scenario complexity index of the candidate offspring. Therefore, all elements in \( C \) are constant. Furthermore, for any state \( i \), it must be transferred to a state in \( Q_{2^n} \). So \( \sum_{j = 1}^{Q_{2^n}} p_{i,j}^C = 1 \) and \( C \) is stochastic.

Let \( M = \{p_{i,j}^M\}_{Q_{2^i} \times \{Q_{2^i} \}} \) denote the transition matrix of the multiple mutation operator, where \( p_{i,j}^M \) denotes the probability that the population is transferred from state \( i \) to \( j \). Similarly, all elements in \( M \) are constant. For the canonical mutation operator, the probability that it transfers a population from state \( i \) to \( j \) is positive [50]. Then, it is known from the pseudocode (Line 4) of the multiple mutation operator that once state \( j \) appears in the candidate offspring, the probability that it is selected as the formal offspring is bigger than 0. Therefore, \( \forall i, j \in \{1, 2, \ldots, \{Q_{2^n} \} \}, p_{i,j}^M > 0 \), and so the transition matrix is positive.

In summary, Theorem 4 is proved by Lemma 3.