FBDScenaGen+: GA-based High-Quality Scenario Generator for FBD Simulation

Eui-Sub Kim[1], Sejin Jung[2], Junbeom Yoo[3], Young Jun Lee[4] and Jang Soo Lee[5]

1. Computer Science and Engineering, Konkuk University, Seoul, Korea (atang34@konkuk.ac.kr)
2. Computer Science and Engineering, Konkuk University, Seoul, Korea (jsjj0728@konkuk.ac.kr)
3. Computer Science and Engineering, Konkuk University, Seoul, Korea (jbyoo@konkuk.ac.kr)
4. I&C.HF Division, KAERI, Daejeon, Korea (yjlee426@kaeri.re.kr)
5. I&C.HF Division, KAERI, Daejeon, Korea (jslee@kaeri.re.kr)

Abstract: Simulation plays an important role in verifying the functionality of FBD programs. When we develop a safety-level PLC-based digital I&Cs in NPP, the simulation quality is important to demonstrate functionally correct/safe. The simulation quality is derived from the scenario quality based on test coverages such as structural code coverage and fault coverage. This paper introduce a tool, FBDScenaGen+, which generates a set of high-quality scenarios based on FBD structural code coverage. We apply genetic algorithm (GA) with the tool in order to increase the quality. The case study will show a feasibility and effectiveness of the proposed method and tools.
Keyword: FBD, Simulation Scenario Generation, Genetic Algorithm

1 Introduction

Function block diagram (FBD)[1] is a widely used programing language for developing a safety-level PLC-based digital I&Cs in NPP. In the PLC-based software development process, the logic is modeled by FBD program in the design phase, then the FBD program is translated into C program in the implementation phase. Verification of FBD program is usually performed by unit testing based on test coverage such as internal structure of the code[2].

Functional verification for assuring the functionality of the system is performed on C program translated from FBD program[3,4]. Therefore, the verification time is often deferred to the implementation phase. To advance the verification time, we developed the FBD simulation framework, which directly reads FBD program and simulates the program.

When simulating the FBD program, scenarios for the simulation are derived from requirements expressed in natural language and manually generated by domain engineers[5]. Even if the scenarios are sufficient to verify all functional requirements, some of the internal design errors may remain the design. Since the scenarios are only for ‘functional’ operation of the design, it is hard to reveal all of the design errors without an exhaustive investigation of the internal structure.

We introduce a high-quality scenario generator, FBDScenaGen+, for FBD simulation. It automatically generates high-quality scenarios based on FBD structural coverage. We use two types of coverage, toggle coverage and MC/DC coverage, we developed[6]. Each coverage check which points in the design are exercised/executed or not. It will play a baseline whether the scenarios have high quality or not and whether we procced the simulation or not.

For high-quality scenario generation, the ‘FBDScenaGen+’ use basic techniques of genetic algorithm (GA). The GA is a search and optimization algorithm inspired by natural selection and natural genetics[7][8]. It solves a problem based on randomized techniques, but it can provide near-optima solutions through an evolution, as the generations went by.

In the paper, we check a feasibility and efficiency of applying GA techniques to the scenario generation. The case study with primarily version of FBD program of RPS BP programs of ARP-1400 in Korean nuclear power plant[9] shows the efficiency of the proposed method and tool. The result is that the quality of scenario is steadily increased from the initial generation to the final generation.

The organization of this paper is as follows: Section 2 and section 3 overview the FBD structural coverage and GA techniques, and the FBD simulation framework and its supporting tools briefly. Section 4 introduce an application of GA techniques to the scenario generation focusing the ‘FBDScenaGen+’. Section 5 briefly looks at case study and experimental result. Section 6 concludes the paper.

2 Background

2.1 FBD Structural Coverage

The FBD structure coverage is a metric for measuring simulation effectiveness that help determine when a system is adequately tested. We proposed two types of FBD structure coverage, toggle coverage and MC/DC coverage. We applied the idea of structural coverage adequacy criteria of software testing, which could apply into a software unit/component with single execution cycle, into the FBD simulation to estimate the structural coverage of FBD simulation scenarios against a whole FBD program with 100 ~ 10,000 execution cycles.

2.1.1 Toggle Coverage for FBD program

Toggle coverage 오류! 참조 원본을 찾을 수 없습니다. is one of the oldest coverage measurements in hardware design. It measures the bits of logic that have toggled during simulation. It focuses on how a Boolean variable is changed and confirms the both 1-to-0 and 0-to-1.

The criterion of the toggle coverage for FBD program is that an output of each Boolean function block and output variable should be toggled forth and back at least once by a simulation scenario. The rationality of the criterion is that a fault in a function block and output variable can only be revealed by executing the faulty function block. If a bit never changes value, it is usually an indication that a mode is not being exercised in the design or a data path has a stuck-at issue.

2.1.2 Block MC/DC Coverage for FBD program

MC/DC (Modified Condition/Decision Coverage) is the most widely-used structural coverage in software testing[11][12]. It tries to effectively test important combinations of conditions, without exponential blowup in test suite size. Important combinations mean that each basic condition shown to independently affect the outcome of each decision. Block MC/DC coverage uses the MC/DC to measure how many important combinations of conditions are covered by a simulation scenario against a function block. The way to measure Block MC/DC on a Boolean function block is the same with MC/DC, but the detail is out of the scope of this paper.

The criterion of block MC/DC coverage for FBD program is that each important condition should be executed at least once by a simulation scenario. The rationality of the criterion is that multiple condition coverage is impractical in practice. MC/DC was developed to achieve many of the benefits of multiple-condition testing while retaining the linear growth in required test cases of condition/decision testing.

The FBD structural coverages are used to evaluate the quality of scenarios and to calculate a fitness function of genetic algorithm.

2.2 Genetic Algorithm

The GA (Genetic Algorithm)오류! 참조 원본을 찾을 수 없습니다.[8] is a search algorithm based on the mechanics of natural selection and natural genetics. It is widely used for finding and solving optimization and complex search problems. Genetic algorithm is part of the evolutionary algorithm category[12]. They are based on the evolutionary ideas of natural selection and genetics which mimics natural phenomenon such as the law of dominance. Genetic algorithm is commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.

2.2.1 Terminology

The basic GA terminologies consist of gene, chromosome and population. The population is a set of chromosomes and each chromosome consists of genes. The relationship is represented in Fig 1. The evolution of population is proceed iteratively, and each iteration we call a generation. The generation start from an initial generation and the evolution is proceeded until a

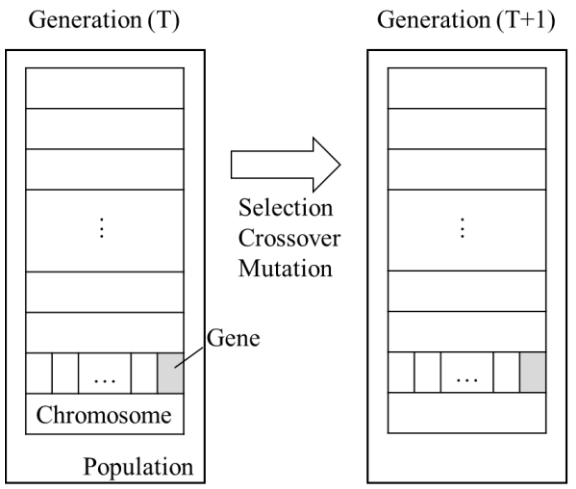

Fig 1. Basic Terminologies of GA

fixed length of generation or the evolution is not improved. In the paper, we consider a generation with a set of scenarios, a chromosome is a scenario, and a gene is a rate of change of value. The quality of scenarios would increase as the generation evolved.

2.2.2 Overall Process

A typical process of GA consists of initialization, selection, crossover and mutation operators. The initialization is to prepare an initial population, which is generally constructed randomly. The selection operator choose a proper portion (chromosome) in the existing population for a new generation. In other words, it performs a filtering process to discard bad genes and preserve good genes in order to find best solution. The crossover operator is to reproduce new children (offspring) by combining genes in the existing chromosomes. The mutation operator has a similar role for the crossover operator in terms of genetic diversity. Contrary to the crossover operator, the mutation operator alters one or more genes in a chromosome for new children. The generation will evolve through the set of operators.

3 FBD Simulation Framework

The FBD simulation framework provides the verification environments for safety-critical digital systems in NPPs. It help engineers verify the functionality of FBD program in the early development phase, design phase. This framework is now embedded in NuDE 2.0 (Nuclear Development Environment), which is a formal method based software development environment[13][14].

The FBD simulation framework[15] consists of various supporting tools such as 'FBDScenaGen', 'FBD Editor'[16], 'FBDSim' and 'FBDCover'[1]. In this pepper, we focus on 'FBDScenGen', which is upgraded for generating high-quality FBD simulation scenarios.

3.1 FBDScenaGen

'FBDScenaGen' generates a number of simulation scenarios accordance with the design logics. It reads an FBD program modeled with the FBD Editor, and produces a set of scenarios for simulation with

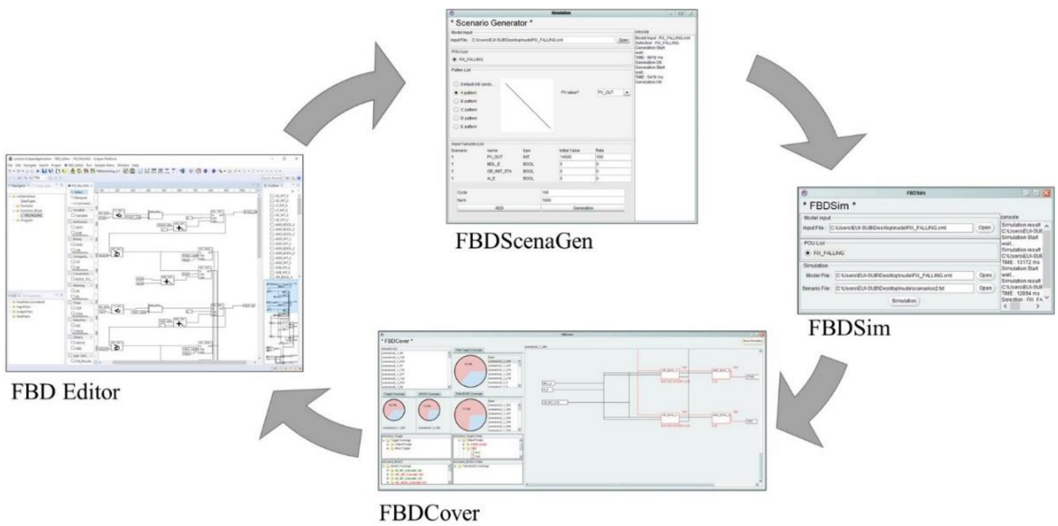

Fig 2. The FBD Simulation Framework

'FBDSim'. The 'FBDScenaGen' is placed in the top side in the Fig 2. We upgraded the 'FBDScenaGen' by applying GA techniques, and we now call the upgraded version as 'FBDScenaGen+'.

3.2 FBD Editor

FBD Editor is an editing tool for FBD program and display the FBD diagram. It is placed in the left side in the Fig 2. It is an independent tool from PLC vendors' SW engineering tools so that it is possible to edit FBD programs complying with the PLCopen TC6 XML scheme. Commercial tools are not compatible with other editing tools for FBD programs.

3.3 FBDSim

'FBDSim' is a simulation tool for FBD program, it reads the FBD program which is written by 'FBD Editor', and generates a set of simulation scenarios for the 'FBDScenaGen'. Simulation is proceeded automatically with the scenarios in batch.

3.4 FBDCover

'FBDCover' is a visualization tool to provide quantitative information of FBD simulation coverages with graphics, as presented in right side in Fig 2. It is executed seamlessly after completion of the FBD simulation by 'FBDSim'. Total coverages and basic coverages are depicted separately, and clicking an individual scenario shows the basic, i.e., showing toggle coverages and block MC/DC coverage each. It also identifies a set of uncovered points in regard with the coverages. SW engineers can use the quantitative information to refine an individual simulation scenario or to improve the total coverages of the whole set of FBD simulation scenarios.

4 FBDScenaGen+: GA-based High-Quality Simulation Scenario Genera -tor

The 'FBDScenaGen+' is an upgraded version of 'FBDScenaGen' for enhancing the quality of FBD simulation scenarios. The original version randomly generates FBD simulation scenarios, whereas the upgraded version generates higher quality scenarios based on FBD structural coverages and genetic algorithm.

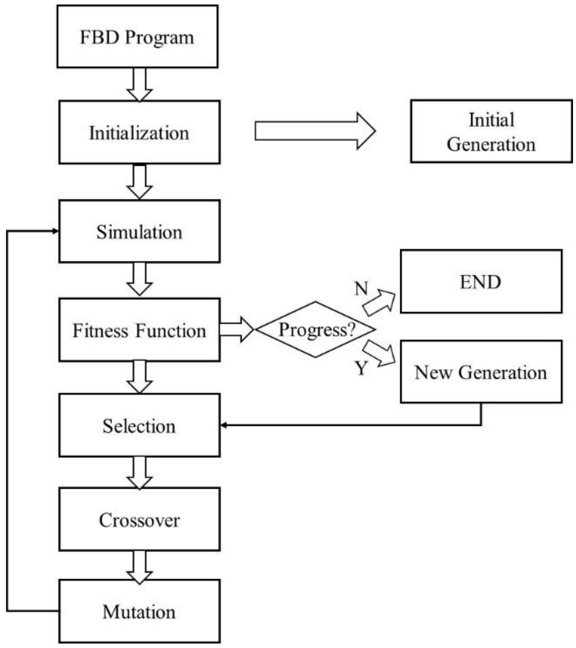

Fig 3 Whole process of 'FBDScenaGen+'

The whole flow of 'FBDScenaGen+' is depicted in Fig 3. It first reads an FBD program, then generates a set of initial scenarios randomly. During the simulation, each scenarios evaluated with fitness function based on FBD structure coverage. If the quality of scenarios reach a certain threshold, it stops an evaluation. On the other hand, if the quality did not reach a threshold, it proceeded three GA operator, selection, crossover and mutation.

3.1 A genetic representation

In genetic algorithms, it is important to represent a solution domain with a genetic representation which should be considered. A chromosome is a set of solutions for solving the domain problem. We define that a chromosome consists of a sequence of increasing and decreasing rate of change. The both of increasing and decreasing rate of change are represented by a gene property in chromosome. It is meaningful to define a gene with a rate of change, because the scenario has a tendency of input sequence and the tendency is very important information in the scenario.

Fig 4 show a genetic representation of simulation scenarios. The arrow represents an increasing and decreasing change, and a value means a gap of change.

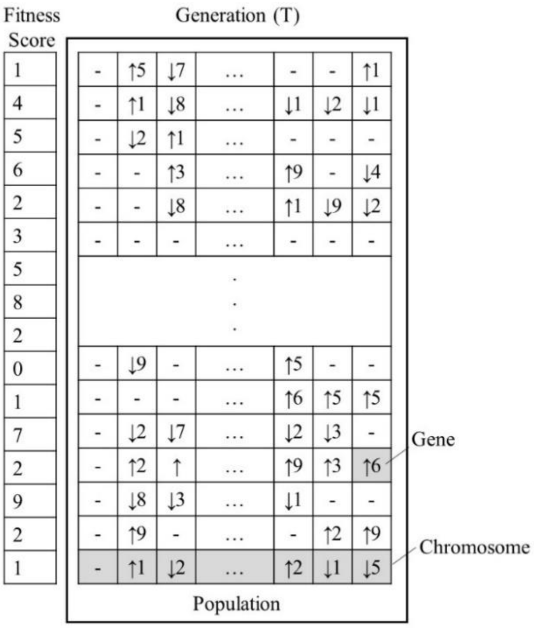

Fig 4. A genetic representation of scenario

3.2 Initialization

The initial population is randomly constructed by the 'FBDScenaGen+'. It produces a number of random simulation scenarios according to the RPS trip (shutdown) logics. It requests auxiliary information on the FBD program in order to make the generated scenario meaningful ones. Initial values and rate of change of all input variables, trip/pretrip set-points, the overall percentage of trip situations, and the number of PLC execution cycles for each scenario are requested.

It produces three basic types of pattern (straight line pattern, slope pattern (rising and falling) and 'V' pattern (rising-falling and falling-rising), which imitate a real RPS input sequence. The shape of the patterns is shown in Fig 5. The intent of first shape of the Fig 5 shows normal operation of the RPS system. A trip or pre-trip signal is not produced such a case. The purpose of the second and third shapes is to produce a trip or pre-trip signal, which is a prime and critical function of the RPS system. The last two shapes are representing patterns about a recovery operation from trip or pre-trip situations. The population size can be defined flexibly, but we generated the fixed-length scenarios (1,000) and the fixed-length cycle of each scenarios (100). We proceeded the evolution until the fixed-length generation (100).

Fig 5. The shape of scenario patterns of the 'FBDScenaGen+'

3.2 Application of Genetic Algorithm

We apply basic techniques of GA for selection, crossover and mutation operations, though there are many advanced techniques. Our primary concern is to check an efficiency and possibility of application of GA techniques to the simulation scenario generation. The following sections introduces which techniques we applied.

3.2.1 Selection

The selection operator is to choose proper parents in the current population for a new generation. Parent selection is very crucial to the convergence rate of the GA as good parents drive individuals to a better and better solutions.

The 'FBDScenaGen+' takes a fitness proportionate selection, roulette wheel selection, with the selection operator. The basic idea of roulette wheel selection is that the better the chromosomes are, the more chances to be selected they have. Fig 6 shows an example of roulette wheel selection. The portion of each chromosome in the whole wheel are defined based on fitness function score. All of chromosomes can be randomly selected, but the higher score chromosome they have, the more portion in the roulette wheel they

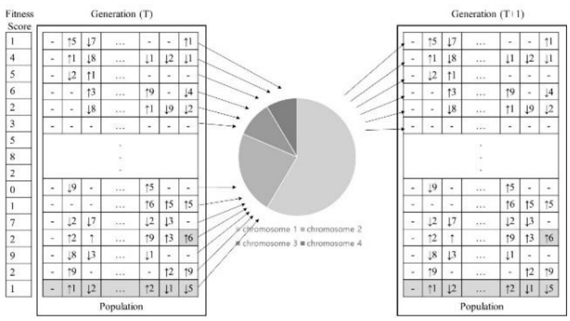

Fig 6. An example of roulette wheel selection

occupy. Hence, they would be selected to higher probability.

We select 50% simulation scenarios from existing population by rolling the wheel. We assign a large portion according to the fitness score that scenarios have. After random selection based on the wheel, the selected scenarios are preserved to the next generation, whereas the others are discard. Since the selection operator is biased toward more highly fit individuals, the fitness of the overall population is expected to increase in successive generations.

3.2.2 Crossover

The crossover operator is to reproduce a new chromosome by combining the existing chromosomes, which is analogous biological crossover. It is basic operator of GA and the performance of GA very depends on the operator.

The 'FBDScenaGen+' takes a single point crossover technique with the crossover operator. The single point crossover randomly selects a separation point and copies two parent's gene sequence to get new children. The children have a swapped gene sequence between the two parent chromosomes. The underlying idea is that the selected parents have a good gene sequence for finding best solution, so that it is worth to utilize the gene sequences. As we consider a gene is an increasing and decreasing rate of change of value, we copy the rate of change to the children chromosome.

We select 50% simulation scenarios for a next generation. The other 50% scenarios are gotten by crossover operators based upon selected scenarios. Fig 7 shows an example of single point crossover. It is important to select a proper separation point, but we did not consider any strategy for selecting the point in this paper. We choose a random based selection point. The aim of the paper is to apply basic GA techniques to the scenario generation for FBD program simulation.

3.2.3 Mutation

The aim of mutation operator is identical for the crossover operator in terms of the genetic diversity. The mutation operator reproduces a chromosome by altering a specific gene of the existing chromosomes, which is analogous biological mutation. The mutation may be defined as a small random tweak in the chromosome. It is basic operator of GA and influence the performance of GA in common with crossover operator.

The 'FBDScenaGen+' takes a swap mutation technique with the mutation operator. The swap mutation operator randomly selects two genes and swaps their position to get new children. The children have a new gene sequence interchanged the random selected two genes. Fig 8 shows an example of swap mutation. The first step is to select a chromosome in the population, randomly, and select two genes for exchanging, then exchange two genes to reproduce a new chromosome.

In the Fig 8, the two yellow boxes are randomly selected genes to exchange the values. Contrary to crossover operator, the mutation operator selects a candidate chromosome in the current population, and reproduces a chromosome for current population. In the Fig 8, the original chromosomes are included in *t+1* generation and the mutated chromosomes are included *t+1* generation, too. It is also important to select which chromosomes and genes are selecting for new children. However, we also do not consider this aspects.

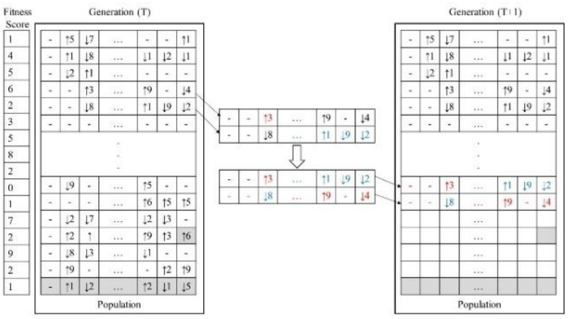

Fig 7. An example of single point crossover

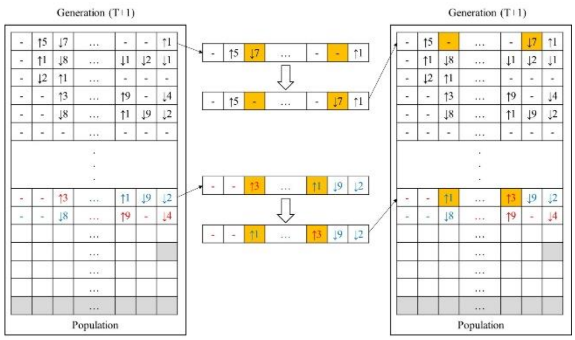

Fig 8. An example of swap mutation

3.3 Fitness function

The fitness function is a function to evaluate and determine how to fit a given solution is in solving the problem. We should come up with the best set of solutions to solve a given problem. The fitness score indicates how close it came to meeting the overall specification of the desired solution. After the basic GA process such as crossover and mutation, the score calculated by the fitness function.

In the paper, a chromosome is a simulation scenario and the fitness function is derived by FBD structural coverages such as toggle and MC/DC coverage. Given FBD program, we generate a set of scenarios and calculate the quality of scenarios based on the structural coverage. After repeated scenario generation including GA operations process and evaluation process with the fitness function, we can obtain a compact set of scenarios, which would high FBD structural coverages.

The below is a fitness function we defined and used:
fitness for toggle coverage:

$$f_T = \frac{\begin{pmatrix} nuber\ of\ toggled\ blocks \\ and\ output\ variables \end{pmatrix}}{\begin{pmatrix} number\ of\ boolean\ blocks \\ and\ output\ variable \end{pmatrix} \times 2}$$

fitness for MC/DC coverage:

$$f_M = \frac{\begin{pmatrix} number\ of\ simulated \\ imporant\ combinations\ of\ conditions \end{pmatrix}}{\begin{pmatrix} all\ important\ combinations\ of\ conditions \\ for\ all\ boolean\ function\ blocks \end{pmatrix}}$$

fitness function:

$$f = f_T \times f_M$$

5 Case Study

5.1 Target system

We performed a case study with primarily version of FBD program of RPS BP programs of ARP-1400 in Korean nuclear power plant in order to identify the efficiency of the proposed method. The BP reads 18 sensor values from a nuclear reactor and decides to generate trip/pre-trip signals out to shutdown the reactor immediately, if any value is out of safe range. It consists of 18 shutdown logics of FBD programs, but we used representative trip logics ‘fixed set-point rising/falling trip’. We will explain the simulation process, experiments results and enhancement strategies in the following chapter.

5.2 Experimental Result

In the initialization, we generated 1,000 initial scenarios and evolved to the 100 generation. During the repetition (evolution of generation), ‘FBDScenaGen+’ proceeded a selection operation (roulette wheel selection), crossover operation (single point crossover) and mutation operation (swap operation) for a new generation. Each scenarios are evaluated by a fitness function based on FBD structural coverages such as toggle and MC/DC coverage.

The final quality of scenarios reach approximately 90%, while the quality of initial scenarios is 55%. The quality is increased steadily during the repetition as shown in Fig 9.

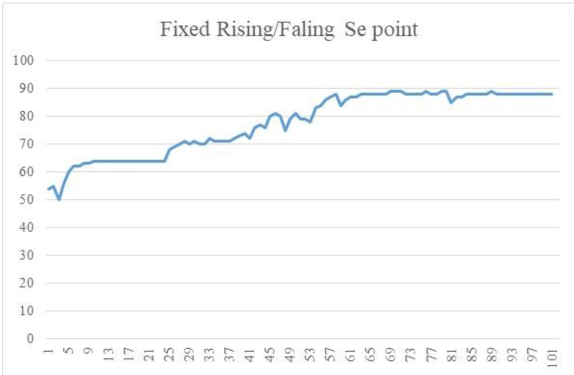

Fig 9. The result of ‘FBDScenaGen+’ by repetition

Some of the results shown a sudden fall in the scenario quality during steadily increasing. The reason is that the crossover and mutation operators accidentally generate some of bad quality scenarios. We should adjust various GA parameters to obtain a good result. Meanwhile, we have confirmed a phenomenon that the dropped quality is shortly recovered during the repetition.

We also found that the mutation operator has a bad effect on the quality of simulation scenarios. The mutation is useful operator in terms of gene diversity. In the simulation domain, however, it is more important a tendency or good flow of input sequence than gene diversity. We first given 3% mutation rate, but the quality went from bad to worse. Hence, we finally decided to give a mutation rate with 0.3%. Hence, we are able to enhance the simulation quality by adjusting GA parameters.

While the 'FBDScenaGen+' produces a set of scenarios, 'FBDCover' shows the fitness score and uncovered points, visually. The 'FBDCover' is a pair tool with the 'FBDScenaGen+'. Some of the points did not covered as depicted in Fig 10. If the GA solution fall in local optimization, the 'FBDScenaGen+' cannot find a global optimization. The local optimization is a critical issue in the GA filed. In case of local optimization, engineers can directly cover those points referencing the uncovered information in 'FBDCover'.

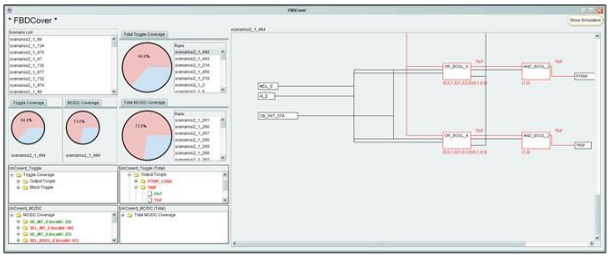

Fig 10. Uncovered points in the 'FBDCover'

5.3 Discussion

In the GA, various parameters are important to obtain good result. For example, we pre-defined that the size of population is 1,000, the number of repetition (evolution) is 100, the selection rate of roulette wheel selection is 50%, and the mutation rate is 0.3%, and so on.

A sufficient number of generations should be well defined to obtain useful combinations for higher quality off-spring. Also, the number of repetition should be well defined, not a 100. The best solution maybe evolve the generation until the fitness score did not be increased. Thus, we are planning to tackle a proper parameter issue to apply various value to the selection, crossover and mutation operator.

6 Conclusions

We applied basic GA techniques to the scenario generation for FBD program simulation in order to generate a high-quality scenarios for FBD simulation. The prime objective of this paper is to check a feasibility and efficiency of applying GA techniques to the scenario generation.

We developed 'FBDScenaGen+', which can automatically generates high-quality scenario for FBD simulation. The scenarios are evolved by GA techniques such as initialization, selection, crossover, mutation. The result, quality of scenarios, is increased during repetition. Although 100% coverage is not accomplished, uncovered points can be ascertained with the 'FBDCover'.

Several issues are remaindered in the 'FBDScenaGen+', for example adjusting of GA parameter. We are planning to considerably research which parameters are precise and accurate. If the efficiency and feasibility are sufficiently revealed, we are planning to adopt high-level techniques such as neural network, machine leaning and so on in the future.

Acknowledgement

This research was supported by the Ministry of Science, ICT & Future Planning. It was also supported by Next-Generation Information Computing Development Program through the NRF funded by the Ministry of Science, ICT (NRF-2017M3C4A7066479) and Basic Science Research Program through the NRF funded by the Ministry of Education (NRF: 2017R1D1A1B03030 065).

References

[1] International Electrotechnical Commission (IEC), International standard for programmable controllers: Programming languages 61131-Part 3, IEC, 1993.
[2] USNRC, Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants, Regulatory Guide 1.171, 1997.
[3] S. Richter, J. Wittig, "Verification and validation process for safety I&C systems," Nuclear Plant Journal, pp.36–40, 2003.

[4] L. Baresi, M. Mauri, A. Monti, M. Pezze, "PLCTools: design, formal validation, and code generation for programmable controllers," IEEE International Conference on System, Man, and Cybernetics, Nashville, USA, Oct. 8–11, 2000.
[5] E. Jee, J. Yoo, S. Cha and D. Bae, "A data flow-based structural testing technique for FBD programs," Information and Software Technology, Vol.51, No.7, pp.1131-1139, 2009.
[6] E.-S. Kim, S. Jung, J. Kim and J. Yoo, "MC/DC and Toggle Coverage Measurement Tool for FBD Program Simulation," Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea, May 12-13, 2016.
[7] Goldberg, David E., Genetic algorithms in search, optimization, and machine learning, 1989, Addison-Wesley, 1989.
[8] Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D., Genetic programming: an introduction (Vol. 1), Morgan Kaufmann, San Francisco, 1998.
[9] Korea Atomic Energy Research Institute (KAERI), Software Design Specification for Reactor Protection System, KNICS-RPS-SD231 Rev.02, 2006.
[10] R. G. Sargent, "Verification and validation of simulation models," in Proceedings of the 37th conference on winter simulation, Florida, USA, Dec. 4-7, 2005.
[11] Radio Technical Commission for Aeronautics (RTCA), Software Considerations in Airborne Systems and Equipment Certification (DO-178B), 1992.
[12] Back, T, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms, Oxford university press, 1996.
[13] J.-H. Lee and J. Yoo, "NuDE: Development Environment for Safety-Critical Software of Nuclear Power Plant," Transactions of the Korean Nuclear Society Spring Meeting 2012, May. 16-18, Jeju, Korea, 2012.
[14] E.-S. Kim, D.-A. Lee, S. Jung, J. Yoo, J.-G. Choi, J.-S. Lee, "NuDE 2.0: A Formal Method-based Software Development, Verification and Safety Analysis Environment for Digital I&Cs in NPPs," Journal of Computing Science and Engineering, Vol.11, No.1, pp.9-23, 2017.
[15] E.-S. Kim, "A technique for demonstrating correctness and safety of program translators: strategy and case study," M.S. thesis, Konkuk University, Seoul, 2015.
[16] D.-A. Lee, E.-S. Kim, Y.-J. Seo, and J. Yoo, "FBDEditor: An FBD Design Program for developing Nuclear Digital I&C Systems," in Korea Conference on Software Engineering (KCSE 2014), Pyeongchang, Korea, Feb. 12-14, 2014.