SOFTWARE TESTING OPTIMIZATION FOR LARGESYSTEMS USING AGENT-BASED AND NSGA-II ALGORITHMS

Muhammad Abid Jamil[1], Mohamed Kidher Nour[1] and
Normi Sham Awang AbuBakar[2]

[1]Department of Computer Science, Umm Al Qura University, Makkah, KSA
[2]Kulliyyah of Information and Communication Technology,
International IslamicUniversity, Malaysia

ABSTRACT

The multiobjective optimization problem is addressed in this article using a novel evolutionary technique tofind a global solution in the Pareto form. The proposed work is innovative because it applies an evolutionary multi-agent system (EMAS) and NSGA-II from various traditional evolutionary methods. The evolution process in NSGA-II and EMAS enables thorough exploration of search space, and the employed crowdsourcing mechanism facilitate the accurate approximation of the entire Pareto frontier. The technique is explained in this article, and report the initiatory experimental findings. The product line or large configurable system needs to set specifications, architecture, reusable components, and shared products to develop the features of new products. To maintain high quality, a thorough testing process is required. Testing is necessary for each product of the large system, each of which has a varied set of features. Consequently, a multi-objective optimization technique can be used to optimize the large system testing process. The performance of a multi-objective Non-Dominated Sorting Genetic Algorithm II (NSGA-II) and evolutionary multi-agent system (EMAS) on Feature Models (FMs) to enhance large Systemtesting is reported in this study.

KEYWORDS

multi-agent systems, multi-objective optimization, large configurable systems, non-Dominated Sorting Genetic Algorithm II

1. INTRODUCTION

The last decade has seen a surge in interest in agent technology, although many of its features are still in development. When systems that employ both the agent and evolutionary perspectives are taken into account, the issues become considerably more challenging. Although developing and using such systems can be challenging, doing so frequently creates new opportunities for resolving challenging issues. This is the scenario when an agent uses an evolutionary algorithm to help realize some of its tasks, such as those related to learning or reasoning [1].

The evolutionary multi-agent system (EMAS) is a case, in which a multi-agent system (MAS) aids evolutionary computation by supplying procedures that permit decentralization using the evolution technique. The core concept of EMAS is the population-level integration of evolutionary processes into MAS. Agents can reproduce (create new agents) and die (be removed from the system), in addition to the typical agent-based system interaction techniques (such as communication) [2].

David C. Wyld et al. (Eds): SOEN, SIPP, PDTCA, ITE, CCSIT, NLPCL, DaKM, BIGML, AISC -2023
pp. 47-52, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131204

The agent's fitness, which is reflected in the quantity of the non-renewable resource known as the life energy it possesses, determines the direction of its activity. The agents with higher energy have more capacity for reproduction, while low-energy individuals have a higher chance of dying. This is how the selection is realized. Thus, unlike (extended) traditional evolutionary computation, EMAS can be viewed as a computational method that makes use of a decentralized evolution model. [3].

Based on this concept, a new evolutionary method for searching for a Pareto-optimal global solution to a multiobjective optimization issue may be proposed. Each agent here provides a feasible solution to a certain optimization challenge. Agents obtain information through communication, enabling the determination of the (non-) domination relation concerning others. Then, subordinate agents provide their dominants with a defined amount of life energy. In this manner, while dominated individuals perish and non-dominated agents obtain more life energy and reproduce. Furthermore, the introduction of the crowd mechanism enables a uniform sampling of the entire frontier [4]. Due to the enormous amount of test cases in large systems, testing becomes difficult. It is a fact that as product variants increase, testing of large systems or product lines becomes more challenging. As a result, system testing grows dramatically along with the implemented products. As a result, it causes the testing of large systems or product lines impractical [5].

The outline of this paper is as follows: The NSGA-II and EMAS algorithms are discussed in Section 2 along with how they are used in search-based software engineering. Section 3 discusses the results and in section 4 conclusions about the research have been described.

2. BACKGROUND

This establishes the foundation for a hybrid evolutionary multi-agent system (HEMAS) and goes into great length about a few selected classic metaheuristics that have served as the foundation for the hybridization of EMAS. The EMAS algorithm's primary phase includes the execution of several metaheuristics satisfying particular conditions. Additionally, these circumstances are also examined, and the conclusions of experiments based on challenging continuous optimization problems serve as examples for all discussions. [15].

Additionally, these circumstances are also examined, and the conclusions of experiments based on challenging continuous optimization problems serve as examples for all discussions. EMAS can be considered a "proactive" complementary to traditional evolutionary computation methods which may eliminate some of the discrepancies between evolutionary metaheuristics and actual evolution. In this approach, the solutions or genotypes are allocated to agents and they are conscious of the variety of options they have to improve their solutions. Agents may congregate and engage in competition or resource exchange [15].

Only the rich agent is permitted to reproduce in the first scenario (similar to the selection process in the evolutionary algorithm); while in the case of the second scenario, certain resources of the rich agent are distributed to the poorer agent.[6]. It is important to note that Michael Vose's proposed research served as an inspiration for the Markov chain-based models that were used to formally demonstrate the soundness of EMAS as a global universal optimizer [7]. Additionally, EMAS has several extensions. For instance, one based on immunology [8] was used to address various single-criteria and multi-criteria problems. The detailed EMAS algorithm is summarized following.

EMAS Algorithm Procedure

```
process Start
initialProgress()
producePopoultation
while ConditionTrue() do
meetingProcess()
reproduce()
deadProcess()
updateProgress()
end While
end process()
```

Non-dominated sorting genetic algorithm II (NSGA-II) was proposed by Deb et al. [9] and has three key characteristics: 1) It makes use of the elitist law; 2) it concentrates non-dominated solutions; and 3) it makes use of the completely developing a preservation mechanism as a quality. The algorithm allows the primary population (Pt) and secondary population (Qt) to be combined in any way that satisfies the domination rule, while F1 only includes non-dominated solutions. The sorting method used by NSGA-II is displayed in Figure 1 [9].

The application of Multiobjective Evolutionary Algorithms (MOEA) in large system testing is discussed in this study. This study looks at how MOEA algorithms can get the right number of test cases to ensure excellent coverage with minimal testing effort. The results of two algorithms EMAS and the Non-Dominated Sorting Genetic Algorithm II are compared in the paper. The application of Multiobjective Evolutionary Algorithms (MOEA) in large system testing is examined in this study.

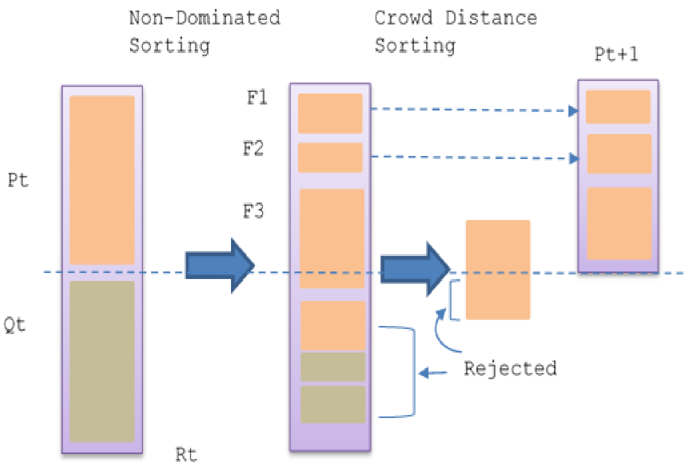

Figure 1: NSGA-II Procedure [9]

The test case selection problem was initially addressed by Yoo et al. [10], who also investigated how the Pareto efficient technique might be used to solve or improve many-objective problems. These search-based algorithms outperformed the greedy technique in four applications from the Siemens suite when the two objectives strategy was used with them [11].

3. EXPERIMENTS AND RESULTS ANALYSIS

The goal of the proposed research is to adapt the EMAS and NSGA-II algorithms to optimize the testing process of the large system feature model. Our technique will lessen the testing exertions for large configurable systems or products. The goal of this research is to generate optimal Pareto fronts using the NSGA-II and EMAS algorithms for feature selection in terms of optimal test cases for large customizable systems.

The two main objectives need to be optimized in our research; 1) minimize the number of products (in terms of reducing the number of test cases) and 2) minimize the testing cost.

Table 1: Attributes for Video Player System

Large System	Features	Configurations	# Pair (using SAT Solver)
Video Player	71	4.5×10^{13}	7528

Table 1 shows the large system feature model Video Player [12] [14] that has been chosen for our research, along with some of its key features including the number of pairs and configurations. The algorithms employed with certain parameter settings in our research are displayed in Table II below. The 200 population size has been chosen for each MOEA algorithm, along with the other parameter values listed in Table II.

Table II: Parameters Adopted

Algorithm	Population	Crossover Operator	Mutation Operator
Video Player	200	60%	30%

Table III: Video Player System Results in Comparison

Algorithms	Solutions	Non-Pareto-Dominance	Pareto-Dominance	% Non-Pareto-Dominance	% Pareto-Dominance
EMAS	219	43	176	19.7	80.3
NSGA-II	187	27	160	14.4	85.6

The findings related to the Pareto dominance and non-Pareto dominance also covered by Jamil and Zitzler et al. are reported in Table III [5] [13]. The selected large system feature model is considered for the experiment. The number of generations has been adjusted to 500, while the population size has been set at 200. The feature model chosen in our scenario was run five times for each algorithm. The Pareto front for the last generation of EMAS is 219 and 187 for NSGA-II. The non-dominated solutions for EMAS are 43 which provides 19.7 % compared to 27 solutions for NSGA-II which is about 14%. The results show EMAS clearly outperforms NSGA-III in two objectives optimization problems.

4. CONCLUSIONS

The optimization process with the multi-objectives sounds intriguing since when there is consideration to improve one object, it might not optimize another. In this section of our research, we discuss the practical utilization of NSGA-II and EMAS to resolve the issues of search-based software testing. In our method, various metrics are used to evaluate the testing objectives. we applied our methodology to a large system described above selected from the feature model repository as mentioned in Table I. In our proposed research, various metrics are used to evaluate the testing objectives. A distinct large system feature model has been adopted from the feature model repository and utilized to validate our methodology. The experiment's output described that the EMAS algorithm generated more optimized results as compared to NSGAII. In addition to our current effort, we intend to adopt the NSGA-III, which has more than two objectives, to address the issues with large systems testing. We will achieve our goals by conducting comparison studies on more extensive product lines.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number: IFP22UQU4320619DSR113.

REFERENCES

[1] J. Liu and H. Qin. Adaptation and learning in animated creatures. In W. L. Johnson and B. Hayes-Roth, editors, Proc. of the 1st Int. Conf. on Autonomous Agents (Agents'97). ACM Press, 1997.

[2] Godzik, M., Dajda, J., Kisiel-Dorohinicki, M., Byrski, A., Rutkowski, L., Orzechowski, P., & Moore, J.H. (2022). Applying autonomous hybrid agent-based computing to difficult optimization problems. Journal of Computational Science, 64, 101858.

[3] Godzik, M., Idzik, M., Pietak, K., Byrski, A., & Kisiel-Dorohinicki, M. (2020, November). Autonomous Hybridization of Agent-Based Computing. In International Conference on Computational Collective Intelligence (pp. 139-151). Springer, Cham.

[4] Rădulescu, R., Mannion, P., Roijers, D. M., & Nowé, A. (2020). Multi-objective multi-agent decision making: a utility-based analysis and survey. Autonomous Agents and Multi-Agent Systems, 34(1), 1-52.

[5] Jamil, M. A., Alhindi, A., Arif, M., Nour, M. K., Abubakar, N. S. A., & Aljabri, T. F. (2019, December). Multiobjective evolutionary algorithms nsga-ii and nsga-iii for software product lines testing optimization. In 2019 IEEE 6th international conference on engineering technologies and applied sciences (ICETAS) (pp. 1-5). IEEE.

[6] Cetnarowicz, K., Kisiel-Dorohinicki, M., & Nawarecki, E. (1996, December). The application of evolution process in multi-agent world to the prediction system. In Proceedings of the Second International Conference on Multi-Agent Systems, ICMAS (Vol. 96, pp. 26-32).

[7] Byrski, A., & Schaefer, R. (2009, May). Formal model for agent-based asynchronous evolutionary computation. In 2009 IEEE Congress on Evolutionary Computation (pp. 78- 85). IEEE.

[8] Byrski, A., & Kisiel-Dorohinicki, M. (2007, May). Agent-based evolutionary and immunological optimization. In International Conference on Computational Science (pp. 928-935). Springer, Berlin, Heidelberg.

[9] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II." IEEE transactions on evolutionary computation, 6(2), 182-197.

[10] Yoo, S., & Harman, M. (2007, July). "Pareto efficient multi-objective test case selection." In Proceedings of the 2007 international symposium on Software testing and analysis (pp. 140-150). ACM.

[11] Yoo, S., Harman, M., & Ur, S. (2011, September). Highly scalable multi objective test suite minimisation using graphics cards. In International Symposium on Search Based Software Engineering (pp. 219-236). Springer, Berlin, Heidelberg.

[12] M. A. Jamil, "Maintenance of software product line using software testing optimization techniques," Ph.D. dissertation, International Islamic University Malaysia, 2020.

[13] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," IEEE Trans. Evol. Comput., vol. 7, no. 2, pp. 117–132, 2003.

[14] Mendonca, Marcilio, Moises Branco, and Donald Cowan. "SPLOT: software product lines online tools." Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications. ACM, 2009.

[15] Socha, K., & Kisiel-Dorohinicki, M. (2002, May). Agent-based evolutionary multiobjective optimisation. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) (Vol. 1, pp. 109-114). IEEE.

AUTHORS

MUHAMMAD ABID JAMIL is a lecturer at Umm AL-Qura University. He received his M.S. degree in Engineering Management Information Systems from the Royal Institute of Technology, Stockholm, Sweden in 2012, he completed his Ph.D. program at the College of Information Communication Technology, International Islamic University, Malaysia. His research interest includes software product line, search-based software engineering, search-based softwaretesting, evolutionary algorithms, and IoT things.

MOHAMED NOUR is an assistant professor at Umm AL-Qura University. He completed his MSc degree at Imperial College London and his Ph.D. from UWE Bristol UK. His research interests include Software Testing, Data mining Interdependency, and Complexity modeling and simulation.

NORMI SHAM AWANG ABU BAKAR obtained her Master's in Real-time Software Engineering from the University of Technology Malaysia. Her Ph.D. is from the Australian National University. Her main research interest is Software Quality Assurance and Metrics. Dr. Normi is an Associate Professor at the Department of Computer Science, Kulliyyah of ICT, International Islamic University Malaysia.