Multi-objective evolutionary optimization of computation-intensive simulations – The case of security control selection

Bernhard Grill[1], Andreas Ekelhart[1], Elmar Kiesling[2], Christian Stummer[3], Christine Strauss[4]

[1] Secure Business Austria, Favoritenstr. 16, 1040 Vienna, Austria
{bgrill,aekelhart}@sba-research.org

[2] Vienna University of Technology, Favoritenstr. 9-11, 1040 Vienna, Austria
elmar.kiesling@tuwien.ac.at

[3] Bielefeld University, Universitaetsstr. 25, 33615 Bielefeld, Germany
christian.stummer@uni-bielefeld.de

[4] University of Vienna, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
christine.strauss@univie.ac.at

1 Motivation

Simulation-based optimization with multiple objectives is a challenging problem that is relevant in a broad range of application domains. A common characteristic of these problems is that they are computationally expensive because (i) the size of the search space is vast for nontrivial problem instances and (ii) the simulation-based evaluation of each candidate solution is runtime-intensive. Multi-objective evolutionary optimization algorithms tackle the first problem and can provide good approximations of the Pareto front even for large problem instances if the evaluation of an individual solution is not too expensive. However, these algorithms typically require a substantial number of computation-intensive evaluations (i.e. simulation runs) before they might converge.

We encountered this issue in the context of a highly relevant practical application within the context of a multi-year research project on analyzing and improving the security of complex information systems. In this project, we combine conceptual modeling of security knowledge, behavioral modeling of threat agents, simulation of attacks, multi-objective evolutionary optimization, and interactive decision support (cf. Figure 1 for an overview). Our model includes a set of security controls (such as firewalls, patch policies, and employee trainings) that can be applied to a modeled target organization's infrastructure. The simulation component evaluates the security of a given system configuration by performing repeatedly step-by-step attacks on the system following heterogeneous attackers' particular objective(s). It delivers metrics such as the expected impact in terms of confidentiality, integrity and availability losses, detected intrusions, and monetary costs. We apply multi-objective evolutionary optimization techniques to determine Pareto-efficient portfolios of security controls based on the simulation outcomes (details can be found in [2]). Initial experiments showed that evaluating a single individual's (i.e., a control portfolio's) fitness based on the outcome of numerous simulation runs may require several seconds.

In order to improve overall optimization runtime performance for this problem, we employ multiple levers. First, selecting an appropriate metaheuristic technique together with carefully tested parameter settings with suitable convergence properties is a crucial task. To this end, we conducted extensive experiments with multiple population-based metaheuristics and parameter settings. Second, due to the stochastic nature of the problem, each candidate solution must be evaluated via multiple simulation replications. In order to reduce the overall computational cost, we can hence reduce the (average) number of required simulation replications for each individual and/or reduce the runtime spent for each replication. Third, we can adapt the optimization to the problem at hand, e.g., by exploiting domain-knowledge on the genotype structure when creating an initial population. In the following, we discuss our ongoing work on approaches to mitigate these issues and identify good solutions in terms of fitness, convergence, and diversity.

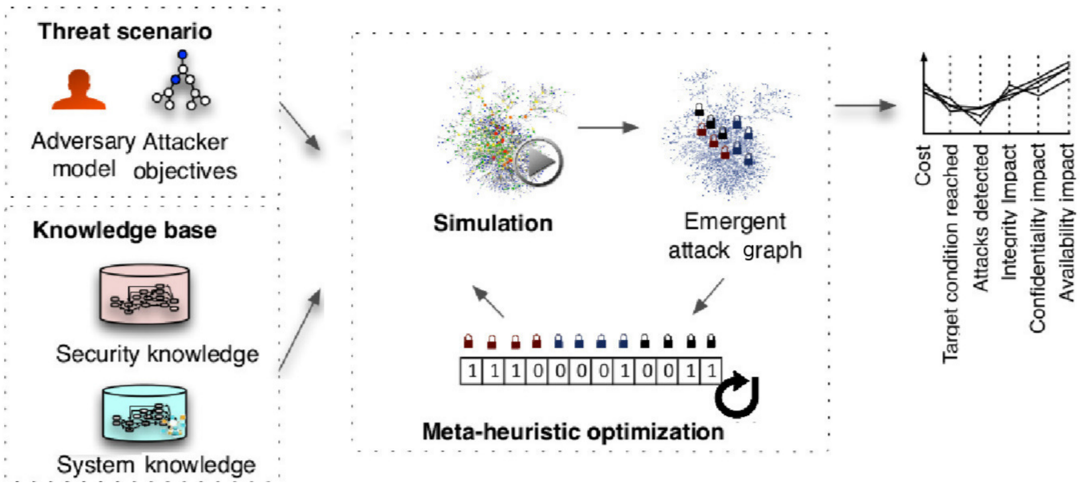

Figure 1: Framework overview for multi-objective simulation-optimization of control sets [2]

2 Improving simulation-optimization performance

Seeding: In order to ensure a high level of diversity, evolutionary algorithms are typically initialized with an initial population that consists entirely of random genotypes. In our initial simulation-optimization experiments, we followed this common practice. However, seeding the initial population with good candidate solutions may yield significantly better solutions with faster convergence. Approaches for systematic population seeding and their respective benefits and disadvantages have been studied extensively for single-objective problems, but the literature on seeding techniques for multi-objective problems is still scarce. We plan to experiment with prior domain-specific knowledge in the creation of an initial population. For instance, the structure of a network may suggest that certain nodes (such as routers connecting internal and external networks) are particularly critical. In that case, adding portfolios which contain security controls that protect these assets to the initial population may be beneficial. A potential disadvantage is that this approach might negatively affect the diversity of solutions.

Genotype structure: Prior domain knowledge about the genotype structure may also be useful for imposing admissibility constraints on the generation of new individuals. Identifying and modeling such domain constraints requires additional effort, but it may significantly reduce the number of evaluated simulations before convergence occurs. In the security domain, such constraints could be imposed based on heuristics or technical requirements (e.g., installing only one anti-virus software per computer).

Caching: Caching is an approach to reduce the runtime of the optimization by eliminating unnecessary, expensive evaluations. Genetic algorithms breed new populations of solutions through crossover and mutation of individuals, which may generate candidate solutions that have already been evaluated in a prior generation. A cache of previously evaluated candidates and their respective objective values, which could be retrieved very efficiently using a hash table or a binary tree, could return already known results. Storage requirements and lookup overheads are moderate. This approach works well for a small number of decision variables [3], but will be less helpful if the search space is very large.

Simulation feedback loop: Utilizing the feedback gained from the simulation may reduce the number of replications performed. Rather than assigning the same number of replications to each individual, in stochastic discrete-event simulations, probability and statistical analysis can be applied to estimate the objective values and adjusting the number of replications assigned dynamically. Besides, the simulation may be stopped whenever a threshold is exceeded indicating that results will be far from acceptable [1].

Parallel metaheuristics: Parallel metaheuristics could ideally speedup runtime almost linearly. This could be achieved by running the simulations in parallel, either on a single machine or in a distributed manner. The corresponding parallelization overhead should be limited and manageable. An important restriction, however, is that the maximum number of computation nodes is limited by the population size as the offspring depends on the results of the previous generation. Talbi et al. describe different parallelization concepts such as cooperating subpopulations or the multi-start approach which could speed up the optimization process tremendously [6]. Furthermore, the parallelization approach could benefit significantly from recent advances in the area of parallel metaheuristics.

Surrogate models: The central idea of surrogate models is to approximate the evaluation procedure with a surrogate model that represents the simulation as accurately as possible, but is substantially less expensive to evaluate. Popular methods to approximate fitness functions include response surface methodologies, Kriging models, and artificial neural networks [5]. We expect neural networks such as multi-layer perceptrons to be particularly suitable, because they excel at capturing complex nonlinear dependencies that are common in our problem domain. Irrespective of the particular surrogate model used, the simulation model would be used to create a set of training data that can then be used to train the surrogate. Once trained, the surrogate model could be used in a number of different ways: (i) If the approximation is sufficiently accurate, the surrogate could replace the expensive simulation-based evaluation entirely during metaheuristic optimization. The final set of efficient solutions obtained may be evaluated again using the actual simulation. (ii) The surrogate model can be used for meta-optimization of the metaheuristic's setup when selecting the most appropriate optimization method and parameter setting for a particular problem instance. Hence, multiple optimization setups could be evaluated using the surrogate model before performing the actual optimization using the original simulation-based evaluation and the identified setup. (iii) The surrogate model could act as a predictor that is integrated into a metaheuristic solution procedure. In this hybrid approach, the surrogate model is used to efficiently predict objective values and act as a filter to select promising individuals that would then be evaluated using the full simulation. A similar prediction approach for single-objective problems was proposed in [4].

Expensive fitness functions, such as simulations, pose a challenge in optimization scenarios. Despite the use of metaheuristic optimization algorithms, simulation-optimization problems hence do not necessarily converge within reasonable runtime. We outlined a number of approaches to tackle this issue and will continue our experiments with these and other techniques. Our goal is to reduce the number of required simulation replications and the runtime spent evaluating each candidate solution in the context of information security control selection. Intermediate results will be presented in our talk. Funding from the Austrian Science Fund (FWF) under project number P23122-N23 is gratefully acknowledged. The research is carried out at Secure Business Austria, a COMET K1 research center supported by FFG. The computational results presented have been achieved using the Vienna Scientific Cluster (VSC).

References

[1] Michael C Fu. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.

[2] Elmar Kiesling, Andreas Ekelhart, Bernhard Grill, Christian Stummer, and Christine Strauss. Evolving secure information systems through attack simulation. In *Proceedings of the 47th Hawaii International Conference on Systems Sciences (HICSS)*, pages 4868–4877, IEEE, 2014.

[3] Jozef Kratica. Improving performances of the genetic algorithm by caching. *Computers and Artificial Intelligence*, 18(3):271–283, 1999.

[4] Manuel Laguna and Rafael Mart. Neural network prediction in a system for optimizing simulations. *IIE Transactions*, 34(3):273–282, 2002.

[5] Soft Computing Home Page. Fitness approximation in evolutionary computation (bibliography), http://www.soft-computing.de/amec_n.html, accessed in March 2015.

[6] El-Ghazali Talbi, Sanaz Mostaghim, Tatsuya Okabe, Hisao Ishibuchi, Günter Rudolph, and Carlos A Coello Coello. Parallel approaches for multiobjective optimization. In *Multiobjective Optimization*, pages 349–372, Springer, 2008.