

THE SIMULATION-BASED MULTI-OBJECTIVE EVOLUTIONARY OPTIMIZATION (SIMEON) FRAMEWORK

Ronald Apriliyanto Halim

Delft University of Technology
Transport, Logistic, Organization
Delft, 2628BX, NETHERLANDS

Mamadou D. Seck

Delft University of Technology
Systems Engineering Group
Delft, 2628BX, NETHERLANDS

ABSTRACT

Simulation and optimization have been successfully combined to solve real-world decision making problems. However, there is no formal structure to define the integration between simulation and optimization. This deters the development of simulation-based optimization methods that have a proper balance between the desired features (i.e. generality, efficiency, high-dimensionality and transparency). This research provides two contributions to the problem above by providing: 1) the design of the framework that facilitates the fulfillment of the aforementioned features; 2) the implementation of the framework in Java. The proposed framework is developed based on Zeigler's modeling and simulation framework and the phases of an optimization study in operations research. The test and evaluation show that the desired features are successfully satisfied.

1 INTRODUCTION

Decision making often involves a trade-off between the satisfaction of multiple conflicting objectives. This can be seen in problem domains covering both social and technological systems, where a single or multiple parties are involved (Ding, Benyoucef, and Xie 2009). Many studies in the field of decision sciences/ operations research have been performed to support decision makers in formulating acceptable solutions to this problem (Hillier and Lieberman 2009). Among those studies, simulation and optimization turned out to be prominent approaches that are widely used to assist decision making process (Hillier and Lieberman 2009; Daalen et al. 2009).

In the simulation-based optimization method, optimization typically functions as the search method that explores the solution space in such a way that solutions leading to the preferred system performance(s), assessed by a simulation model, can be found. Unlike the traditional optimization methods, which use a mathematical model, the simulation-based optimization method allows an accurate representation of the dynamics and stochastic nature of the real system. Furthermore, the best solutions can be found without the tedious effort of manually traversing the whole possible decision alternatives.

There is, however, a knowledge gap in this widely applied method. There has not yet been a formal and detailed explanation of the way simulation and optimization techniques should be integrated. Often, simple conceptualizations of either the simulation or the optimization components are presented as black boxes. This knowledge gap thus, might deter the effective use of optimization techniques by simulation practitioners and vice versa.

Furthermore, other challenges that result from the lack of a structured approach can also be found by looking at the conflicts found in the fulfillment of the broader requirements formulated in Fu (2002). These requirements include generality, transparency to user, high dimensionality and efficiency. Among those requirements, generality very often conflicts with efficiency (Fu 2002; Fu et al. 2000). This can also be seen from the optimization routines that are being developed in this field. Most of them employ evolu-

tionary-based algorithms, which are designed in tight coupling with the modeled-problem to allow efficient exploitation of the problem structure. This tight coupling sacrifices the generality of the method, which would actually allow solving a wide range of problems across different domains (Ding, Benyoucef, and Xie 2006, 2009). While usability can often be facilitated with a user friendly user interface, there are many shortcomings pertaining to efficiency and high dimensionality in the simulation-based optimization commercial packages. A review of two popular optimization routines in commercial simulation software (AutoStat and OptQuest) reveals that both packages lack an efficient multi-objective optimization routine (Fu 2002). AutoStat approaches the multi-objective optimization problem using the classical method in which multiple objectives are aggregated to form a single objective using a weight-vector. The drawback of this method is that for different preferences of the decision maker(s), different weight-vectors have to be used and the same problem has to be solved repetitively. A similar approach was also employed by OptQuest, which only started to provide an efficient multi-objective optimization routine, allowing the analysis of the Pareto Frontier, after the release of engine v 6.5 (<http://www.opttek.com/Products/Documentation.html>).

The framework of modeling and simulation proposed in Zeigler, Praehofer, and Kim (2000), due to its clear characterization of the relevant concepts, is a valuable starting point for tackling the problems mentioned above. In this framework, the principle of separation of concerns is well applied. Firstly, there is a separation between the model and the context under which it is experimented with. Furthermore, there is a separation between the simulator and the simulation model. This framework is therefore a good basis for the detailed formalization of the structural relationships between simulation and optimization techniques. More importantly, the resulting framework can also be used to ensure the proper balance in realizing the various desired features mentioned above.

The proposed framework will make two contributions. The first is to provide a transparent structure and the formal definitions of the simulation-based optimization method. These structure and definitions will contribute to the development of a generic simulation-based optimization methodology. The second contribution is to provide a detailed design and a Java implementation of simulation-based optimization method which is able to fulfill the desired features mentioned in Fu (2002).

The rest of the paper is organized as follows. Section 2 delineates the relevant theories for the development of a simulation-based multi-objective optimization framework. This is followed by the detailed requirements and conceptualization of the proposed framework in Section 3. Next, Section 4 presents the translation of the conceptualization into the system architecture and the detailed designs of the framework. In Section 5, the results of the tests conducted to the framework are presented. Finally, Section 6 presents the conclusion and the recommendation for the future work.

2 SIMULATION-BASED OPTIMIZATION METHODOLOGY

In this section the relevant work both from simulation and optimization fields are presented to provide the theoretical foundations with which a simulation-based multi-objective optimization framework can be achieved. We first present the detailed component of modeling and simulation framework presented by Zeigler in Section 2.1. Subsequently, the description of multi-objective optimization problem and algorithm is presented in Section 2.2.

2.1 Experimental Frame (EF) in Zeigler's Modeling and Simulation Framework

A relevant concept in Zeigler's framework is the experimental frame which realizes the separation of concerns between the model and any data gathering (e.g., statistical measurements), and any control efforts (e.g., starting and stopping of the simulation) that are not performed in the real system. To serve these functions, the concept of experimental frame is formalized with three components which govern the experimentation on a model: the generator, transducer and acceptor (Traoré and Muzy 2006; Zeigler, Praehofer, and Kim 2000). The generator component produces the input segments to the simulation model. Those input segments are then processed by the model using certain state-transition functions to calcu-

late all the resulting state changes. Next, the transducer maps the output variables into outcome measures of interest. Finally the acceptor component monitors the validity of the experiments by checking whether the values of the outcome measures of interest violate the pre-defined constraints of the experimentation.

Furthermore, in a simulation-based optimization method, experimentations can be performed to evaluate the effect of certain alternatives defined by the model builder in respect to certain pre-defined goals. In this case, the simulation model is conditioned to a certain set of circumstances that is defined in a so-called treatment (Daalen et al. 2009). Thus, in an experiment, a specific treatment is given to the simulation model, and the model is run to produce the outcome measures of interest through the acceptor (Figure 1). In the effort of optimizing the system performances, normally a number of treatments that involve different collections of input data are specified and executed in the experiments. However, this can be very challenging and sometimes impossible to do manually if the model is too complex or the relationship between the input data (or often called decision variables in optimization study) and the outcomes measure of interest is not understandable. Therefore, we can see that a computer-based approach that implements an optimization technique is needed to perform the search of the optimal values of the decision variables in respect to the pre-defined goals.

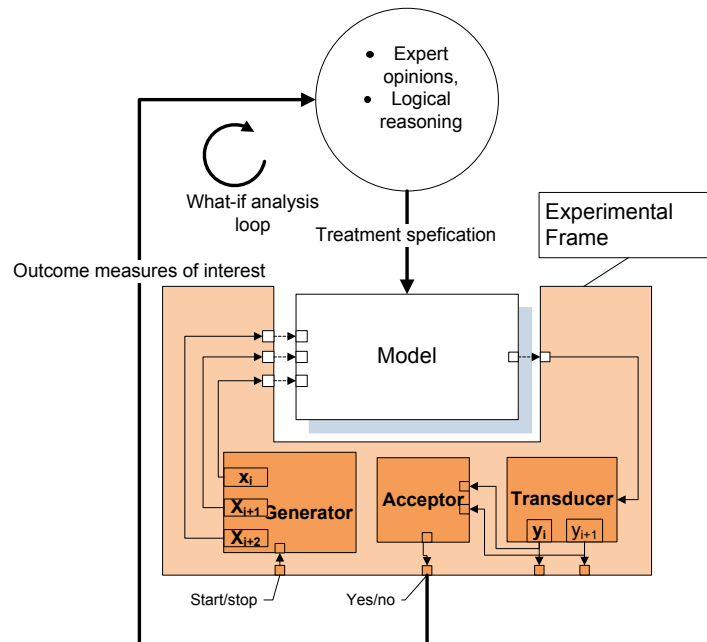


Figure 1: Experimentation process

2.2 Multi-Objective Optimization Methodology

One of the desired features for a simulation-based optimization method is high-dimensionality. This feature mainly comes from the need to solve multi-objective optimization problem. Following section describes the formal structure of such problem and state-of the art algorithm to solve such problem.

2.2.1 Multi Objective Optimization Problem (MOOP)

Following is the formulation of a multi-objective optimization problem:

$$\text{Minimize } [z_1 = f_1(\mathbf{x}), z_2 = f_2(\mathbf{x}), \dots, z_q = f_q(\mathbf{x})] \quad (1)$$

$$\text{Subject to } g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, m \quad (2)$$

$$\mathbf{x} \geq 0 \quad (3)$$

$f(\mathbf{x})$ is the objective function, where $\mathbf{x} \in \mathbb{R}^n$ is a vector of n decision variables, and $g_i(\mathbf{x})$ are inequality constraint that consists of m functions that shape the feasible area.

In single objective optimization, the search process is focused on finding one best solution that is superior to all other solutions. In the case where there are multiple objectives to optimize, it is not always the case that there exists a solution that is optimal in terms of all the objectives due to *incommensurability* and *conflict* among the objectives (Gen, Cheng, and Lin 2008). In case there is conflict among the objectives, a solution that is optimum in one objective may be the worst for the other objectives. In this condition, there is normally a set of solutions that cannot be compared with each other without adding additional information (such as preference structure upon the objectives). These kinds of solutions are normally regarded as *non-dominated* solutions or *Pareto optimal solutions*. Non-dominated solutions have the characteristic that their optimality cannot be improved further without sacrificing at least one of the other objective functions.

2.2.2 Multi-Objective Evolutionary Optimization Algorithm

Since Evolutionary Algorithms (EAs) were introduced to solve MOOP, its use and popularity has increased over the past decade opening the growing research field of Multi-Objective Evolutionary Algorithms (MOEA) (Deb 2008). The main advantage of this method in comparison with the classical methods is that it uses a population based approach in which multiple solutions are simultaneously generated in each of its iteration. This gives MOEA better capability to explore a larger criterion space in shorter computational time. Some of the well-known properties of MOEA are:

- MOEAs have the flexibility to adapt to different problem structures and thus have wide application fields
- MOEAs do not require gradient or derivative values such as what is needed by gradient-based algorithms
- The decision maker(s) does not need to have an a-priori articulation of preferences regarding the accomplishment of all the objectives before the solutions/alternatives are presented

Among, many MOEAs, the proposed framework will use Non-Dominated Sorting Genetic Algorithm II as its optimization engine. It is one of the state-of-the-art methods to solve MOOP (Deb et al. 2002). Like any other EAs, this algorithm also finds its root in evolutionary theory and is built on top of classic Genetic Algorithm. Despite many improvements and additional features that NSGAII has, the fundamental theory that underlies the adaptive capability of this algorithm is the same as that used in standard genetic algorithms. However, there are some features that distinguish NSGA-II from other EAs:

- It employs the elitism principle
- It has an explicit diversity preserving mechanism
- It focuses on finding the non-dominated solutions

We henceforth refer to the proposed framework as the SIMulation-based Multi-objective Evolutionary OptimizationN (SIMEON) framework.

3 SIMEON DEFINITION

To ensure that the conceptualization of SIMEON takes into account all the desired features formulated by Fu (2002), we firstly present the technical requirements elicited based on the real-world needs for the framework in section 3.1. Next, in section 3.2 the conceptualization of SIMEON will be presented based on these requirements.

3.1 SIMEON Requirements

Fu (2002) has formulated four high level features that are desirable in a simulation-based optimization method. In this section we further detail those requirements based on the interviews of the potential users of SIMEON in the real-world.

3.1.1 Generality

Among all the desired features mentioned in Fu (2002), generality is normally of the highest value for both scientific and business worlds as it enables practical implementation of the optimization methods to different simulation problems (Fu 2002; Fu et al. 2000). Consequently, this makes a framework that employs problem-dependent optimization techniques less desirable. Since SIMEON is going to be applied to different problem domains, the optimizer used should be problem-structure independent.

3.1.2 Efficiency

It is important to guarantee the convergence of the optimizer in a reasonable amount of computational time. This is because the framework will be implemented and used in a common office notebook. However, the requirement on efficiency is inevitably affected adversely by the requirement of generality. One consequence of treating the simulation model as a black box (as what is done by metaheuristic) is that the optimizer does not make use of the information regarding the problem structure (e.g., gradient information) to solve the optimization problems (Fu 2002). This leads to a relatively slow performance in comparison to the optimizers that use such information to solve the same problem. However, this condition also can be mitigated by ensuring the extendibility of SIMEON with algorithms that could make use the information regarding the structure of the simulation model such as the perturbation analysis (Fu and Hu 1997), weak derivatives (Pflug 1996), etc.

3.1.3 High-Dimensionality

It is a requirement that gives SIMEON a distinction in comparison to the other simulation-based optimization framework. This requirement mainly comes from the need to optimize multi-objective optimization problems which are ubiquitous in the real world problems. Furthermore, there are also needs that come from the modeling perspective, they are:

- The need to optimize quantitative or continuous or real-valued variables as this need still dominates big number of optimization problems (Ding, Benyoucef, and Xie 2009).
- The need to optimize qualitative or discrete or integer variables which can be used to represent non-quantitative variables including structural alternatives for a simulation model (Azadivar and Tompkins 1999).
- The need to optimize both qualitative and quantitative variables simultaneously (Ding, Benyoucef, and Xie 2006).

3.1.4 Usability

It is important to distinguish and define the users of SIMEON. The first type of user is defined as non-expert users, i.e., those who have at least the basic knowledge of modeling and simulation techniques. These non-expert users are expected to be able to develop simulation models in various simulation packages and use them to carry out an analysis by setting up experiments. On the other hand, the expert users are expected to be the simulation experts who have knowledge level and skills that enable them to develop object-oriented simulation models and use them to solve different kinds of problems.

Another additional requirement would be that both the expert and non-expert users should have the basic knowledge of optimization which enables them to apply optimization techniques appropriately to solve different decision problems.

3.2 SIMEON conceptual design

To systematically perform the integration of NSGAI into the Zeigler's framework, we propose to use the steps of operations research as below.

Step 1. Define the problem of interest and gather relevant data

The modeler has to identify the goals/objectives, the constraints and the relations between all relevant elements of the system. A distinction is made between the controllable variables and the environmental variables.

Step 2. Development of the simulation model

A standard modeling approach as suggested in Daalen et al. (2009) can be used with the following requirements.

- The models should be amenable to coupling with an external device which can modify inputs and measure outputs.
- There has to be a separation of concerns between the problem to solve and the algorithm to use, as a consequence of generality feature.

We adopted the following definitions for the entities related with the model.

- **Treatment.** A treatment is as set of conditions imposed on the model to perform the simulation (Daalen et al. 2009). Such conditions can include input data, initializations, run control conditions, warm-up time, etc.
- **Experimental frame.** The experimental frame is understood as in Traoré and Muzy (2006), with a generator, an acceptor and transducer.
- **Simulation-based optimization experiment.** A simulation-based optimization experiment is an execution of the simulation model with a specific treatment and environmental conditions to produce the outcome measures of interest. Therefore, this type of experiment includes the specification of Zeigler's experimental frame.
- **Simulation-based optimization problem.** A simulation-based optimization problem is a form of optimization problem in which simulation experiments are used to find the optimal values for the decision variables. Thus, simulation-based experiments should be defined within such a problem. Moreover, the specification of this problem should distinguish the decision variables and the other variables (environmental variables) within the simulation model.

Step 3. Develop a computer-based procedure for deriving solutions to the problem from the model.

The jMetal library (Durillo et al. 2006) is used on the optimization side. On the other hand, the DSOL library (Jacobs, Lang, and Verbraeck 2002) is used on the simulation side. Using the routines of NSGAI, SIMEON provides an automatic and iterative specification of the values of the decision variables in such a way they are optimized according to the pre-defined objective functions. Figure 2 depicts the conceptual diagram of the SIMEON framework.

4 SIMEON DESIGN AND DEVELOPMENT

4.1 SIMEON system architecture

SIMEON's architecture is coherent with that of a standard decision support system as proposed in (Burstein and Holsapple 2008). As can be seen later in Figure 3, three layers can be distinguished.

1. *The presentation layer*: This layer serves mainly as the user interface through which the user interacts with SIMEON. There are several interfaces that take the messages from the user into SIMEON, serving as the language systems. On the other hand, there are also interfaces that emit messages to the user, serving as the presentation systems.
2. *The Problem Processing layer*: This layer provides the problem solving services. The optimization engine consists of multi-objective evolutionary algorithm and simulation-based optimization problem objects. Next, the simulation engine consists of the simulation model, the treatment that contains the decision variables for the model, and the simulation-based optimization experiment objects. There can be multiple treatments and simulation-based optimization experiments.
3. *Knowledge/data layer*: a layer where the knowledge system is positioned. This is the layer where all relevant knowledge and information needed by SIMEON are stored. An example of a knowledge system can be independent database systems (e.g. Microsoft Excel, Access, ERP, MySQL, etc) that contain data needed for the simulation or a configuration file for SIMEON.

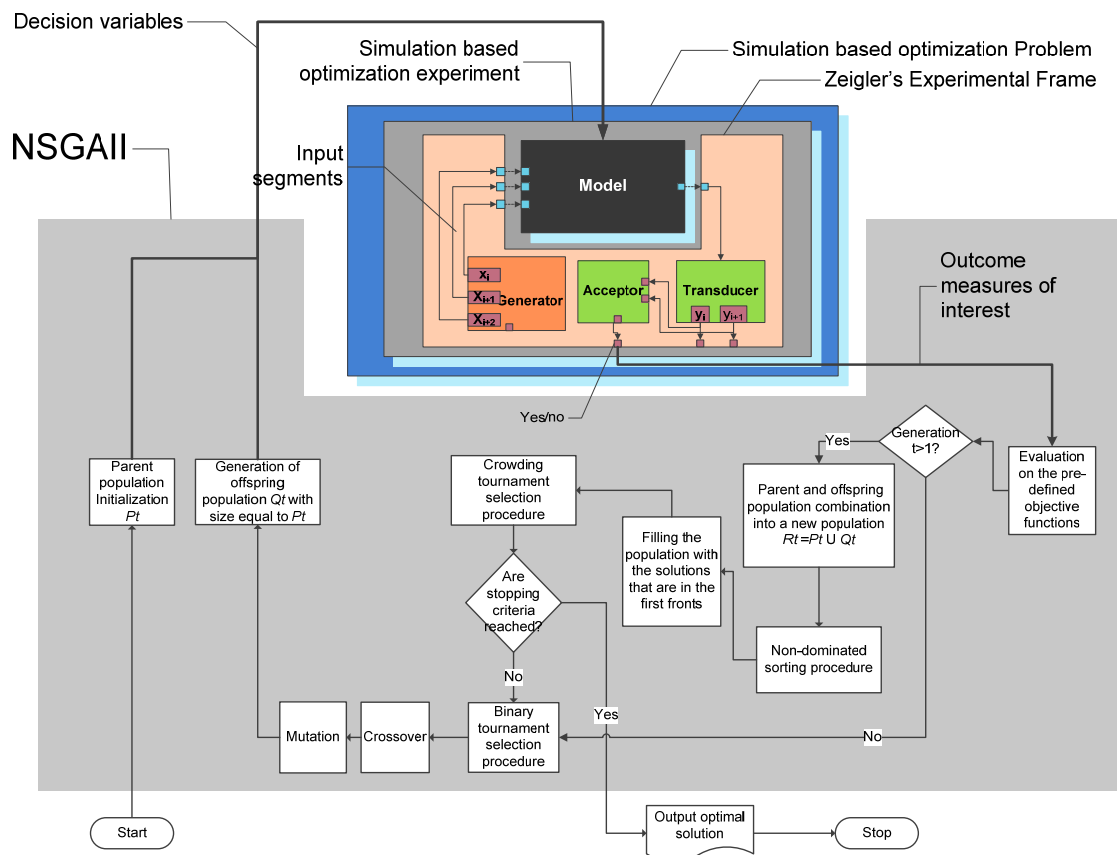


Figure 2: Conceptual design of the SIMEON framework

4.2 SIMEON detailed design

In this section the detailed designs of the SIMEON sub-systems are presented. We first present the implementation structure of the framework in an object oriented language. Next, the design of genetic representations and operators used within SIMEON are shown. Last but not least, the features of the user interface are also presented.

4.2.1 Design of the SIMEON framework

As shown in the system architecture diagram (Figure 3), there are six main components of the simulation and optimization engines. We implement these components as Java based objects.

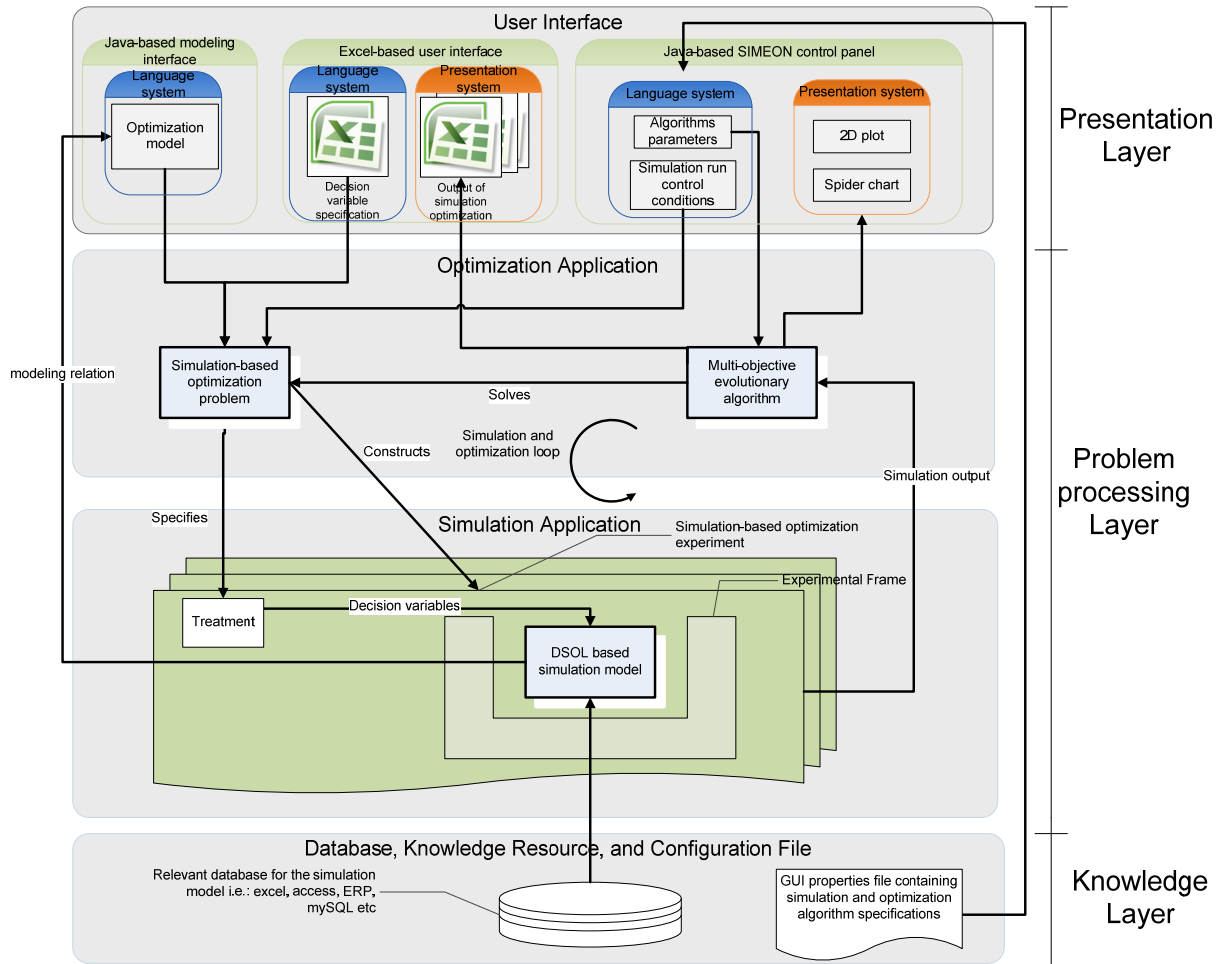


Figure 3: System architecture of SIMEON

The SimOptNSGAI implements the main optimization routine, which solves a Simulation-BasedProblem. The latter defines the SimBasedOptExperiment, which uses the Simulator, Treatment and SimulationOptimizationModel to run unique experiments based on the specifications given to these three objects. Recall that in an execution of the algorithm, one objective function evaluation requires the execution of one simulation experiment. This way, the number of the SimBasedOptExperiment instances created is going to be equal to the maximum number of evaluations that the algorithm is to execute, while there is only one SimulationBasedProblem object. Figure 4 provides an overview of the framework.

4.2.2 Design of genetic representation and operators

Based on the high-dimensionality requirement, two types of decision variables have been identified: quantitative and qualitative decision variables. A simulation-based problem/model might contain quantitative, qualitative or both types of decision variables. To enable SIMEON to have the desired generality and yet also have the ability to solve the problems efficiently, different genetic representations and opera-

tors are implemented. The central idea here is to enable SIMEON users to select the most efficient representations and operators based on the types of the decision problems that are encountered. SIMEON implements different representations including the binary representation, real representation, integer representation, and integer-real representation. The latter form of solution encoding makes use of integer and real representations simultaneously. It is particularly useful to solve simulation-based or mathematical model-based problems that contain both quantitative and qualitative decision variables. To perform crossover and mutation for this representation, two operators are specially implemented: Integer-Real crossover and Integer-Real mutation. These operators perform different genetic operations on different segments of the chromosome depending on the types of the decision variables. The crossover operator combines single point crossover and simulated binary crossover while the mutation operator combines bit flip mutation and polynomial mutation. Figure 5 illustrates this representation.

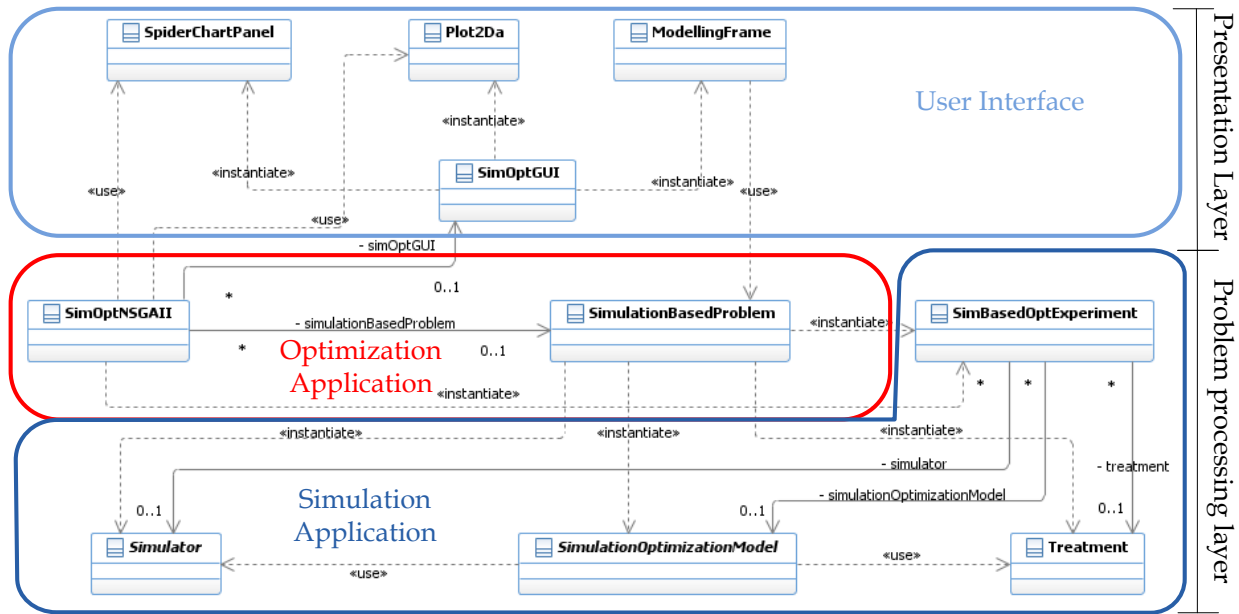


Figure 4: High level UML class diagram of SIMEON framework

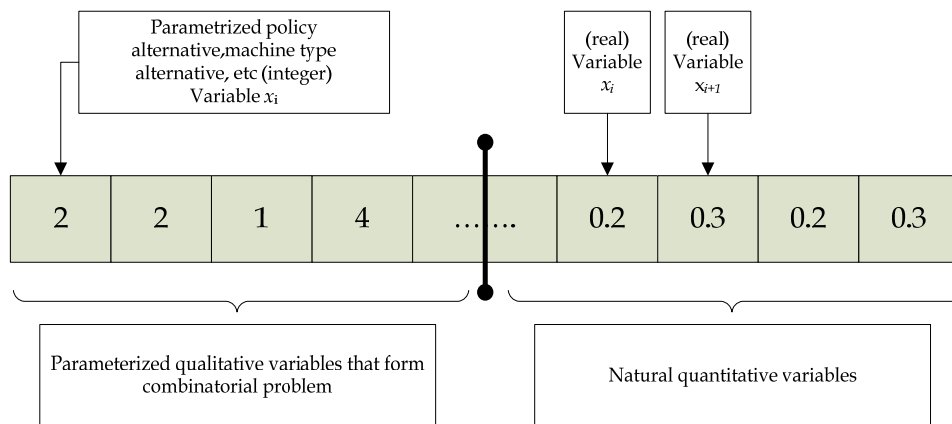


Figure 5: Integer-real representation

4.2.3 Design of the SIMEON User Interface (SIMEON UI)

User interface of SIMEON is the main constituent of the presentation layer. Its design is vital for the usability of SIMEON for both expert and non-expert users. The UI implements two features, a control panel and a modeling interface. For space reasons, we'll only describe the control panel.

Control panel has two main functions:

- Allowing the user to configure the algorithm and the problem.
- Visualizing the non-dominated solutions for either 2 objective functions (on the 2 objective functions plot) or multiple objective functions (on the spiderchart plot).

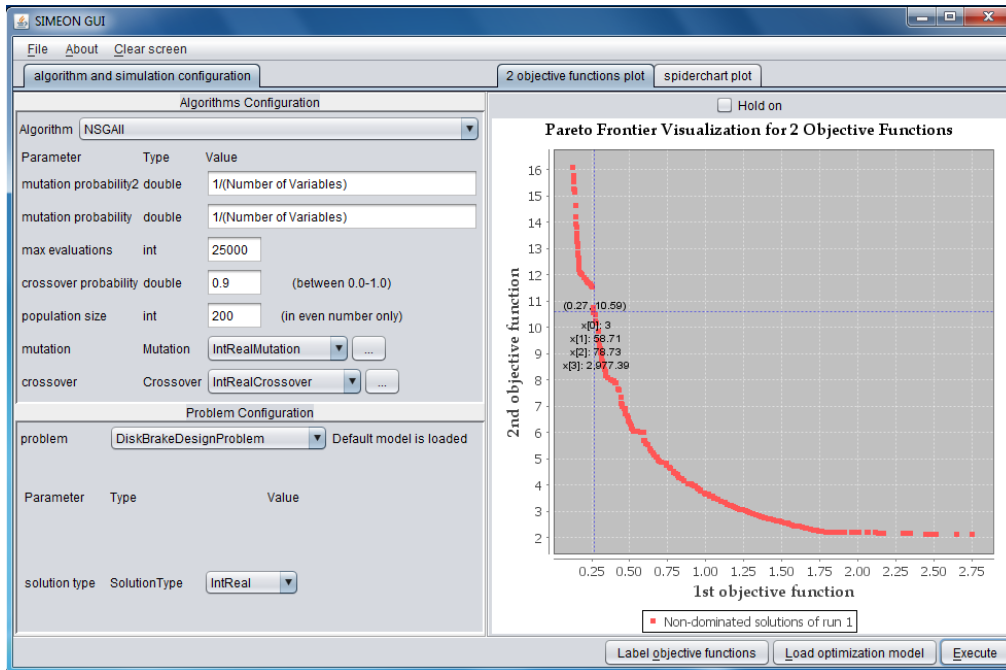


Figure 6: SIMEON control panel

On the top-left panel, the user gets to configure the algorithm to use and its parameters. Next, on the bottom-left panel, the simulation-based problem to solve can be selected from a dropdown menu. The run-control conditions can be specified in the same panel. The right panel is used to show the non-dominated solutions for two objective functions.

5 SIMEON OPERATIONAL TEST AND ASSESSMENT

For the test case, we use a population size of 100; 10,000 evaluations; a 0.9 crossover rate, 1/number of variables mutation rate, and 20 distribution index (for the polynomial mutation and simulated binary crossover) to perform the optimization. Furthermore, to deal with the randomness within the model, each of the solutions generated by the algorithm is simulated for 5 times and the average values of the objective functions are used for further iterations.

5.1 Supply Chain Optimization Problem

The supply chain optimization problem is characterized by the presence of multiple actors /companies and the dynamic interactions that take place during the distribution of products, money and information between those actors. In the simulation model used as a test case, there are two suppliers, one manufacturer, three distributors and four market spots/retailers. Figure 7 illustrates this description.

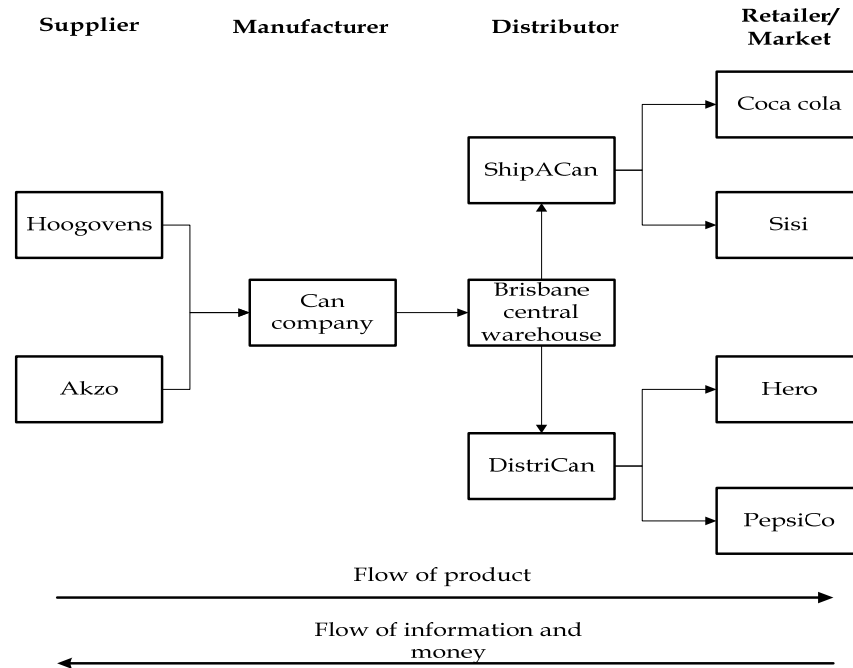


Figure 7: Supply chain network with multiple actors

The supply chain process is started when the four customers (Coca Cola, Sisi, Hero and PepsiCo) generate demands that follow certain statistical distribution. These customers are served by two distributors (DistriCan and ShipACan) as in Figure 8. Various transport modes result in different delivery speeds between those actors. Products are produced by a manufacturer (can company) by using materials supplied by two suppliers (Hoogovens and Akzo). The products are first accumulated in a consolidated central warehouse in Brisbane before they are transported to the distributors.

The objectives are to maximize inventory unit fill rate and to minimize the supply chain cost of a distributor (DistriCan) for a 100 days period. Figure 8 shows the non-dominated solutions of the problem.

We can see that there are two non-dominated solutions that can be selected for a decision. The first solution suggests setting the service level rather high (96%) while the second solution suggests a lower service level (87%). The first solution, however, indicates a good performance on the inventory unit fill rate (93%) with relatively higher cost (28 cost unit) while the second solution gives lower cost of the supply chain (22 cost unit) with relatively lower inventory unit fill rate (89%). Given this information, the decision maker(s), can have the assurance of picking a solution out of the best alternatives possible. Further process of decision making may use additional preferences of the decision maker(s) depending on their specific target or circumstances.

6 CONCLUSIONS AND FUTURE RESEARCH

In this research we have identified the knowledge gap in the integration of simulation and optimization techniques. Furthermore, we also presented the challenges in developing a simulation-based optimization method that fulfills generality, efficiency, high-dimensionality, and usability features.

To address those problems, we design and implement SIMEON, a framework designed based on the aforementioned features. The conceptual design of the SIMEON framework has successfully provided the answer for the knowledge gap. We presented the definitions and the structure that make clear the way optimization and simulation techniques can be integrated in a theoretically sound framework.

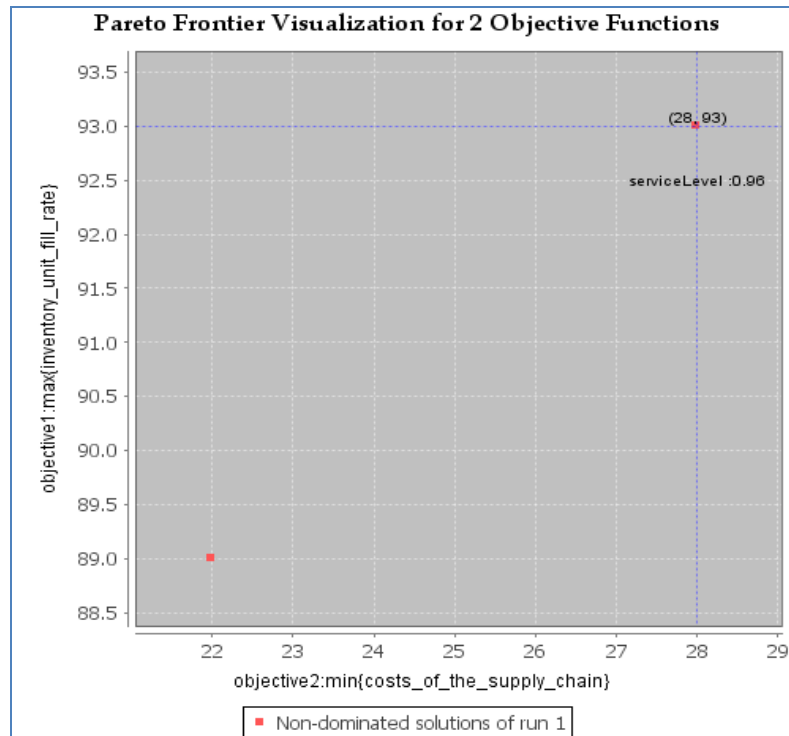


Figure 8: Pareto front for supply chain optimization problem

Next, based on the conceptual design, we presented the system architecture and the detailed design of SIMEON. Two cases are used to test the performance of SIMEON. The results show that the SIMEON framework is able to realize the desired features.

As for the future research, we plan to further develop SIMEON with different evolutionary-based algorithms and perform further testing with real-world and benchmark problems.

REFERENCES

- Azadivar, F., and G. Tompkins. 1999. "Simulation Optimization with Qualitative Variables and Structural Model Changes: A Genetic Algorithm Approach." *European Journal of Operational Research* 113(1):169-182.
- Burstein, F., and C. Holsapple. 2008. "DSS Architecture and Types." In *Handbook on Decision Support Systems I*, edited by F. Burstein and C. Holsapple. Berlin, Heidelberg: Springer.
- Daalen, C. van, A. Verbraeck, W. Thissen, and P. Bots. 2009. "Methods for the Modeling and Analysis of Alternatives." In *Handbook of Systems Engineering and Management*, 2nd edition, edited by A. P. Sage and W. B. Rouse, 1127-1169. New York: John Wiley and Sons.
- Deb, K. 2008. "Introduction to Evolutionary Multiobjective Optimization." In *Multiobjective Optimization*, edited by J. Branke, K. Deb, K. Miettinen and R. Slowinski, 59-96. Berlin, Heidelberg: Springer
- Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6(2):182-197.
- Ding, H., L. Benyoucef, and X. Xie. 2006. "A Simulation-Based Multi-Objective Genetic Algorithm Approach for Networked Enterprises Optimization." *Engineering Applications of Artificial Intelligence* 19(6):609-623.

- Ding, H., L. Benyoucef, and X. Xie. 2009. "Stochastic Multi-Objective Production-Distribution Network Design Using Simulation-Based Optimization." *International Journal of Production Research* 47(2):479 - 505.
- Durillo, J. J., A. J. Nebro, F. Luna, B. Dorronsoro, and E. Alba. 2006. "jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics." Departamento de Lenguajes y Ciencias de la Computacion, University of Malaga.
- Fu, M. C. 2002. "Feature Article: Optimization for simulation: Theory vs. Practice." *INFORMS J. on Computing* 14(3):192-215.
- Fu, M. C., S. Andradottir, J. S. Carson, F. Glover, C. R. Harrell, Y.-C. Ho, J. P. Kelly, and S. M. Robinson. 2000. "Integrating Optimization and Simulation: Research and Practice." In *Proceedings of the 2000 Winter Simulation Conference*, edited by J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 610-616. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Fu, M. C., and J. Q. Hu. 1997. *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. New York: Springer-Verlag.
- Gen, M., R. Cheng, and L. Lin. 2008. *Network Models and Optimization - Multiobjective Genetic Algorithm Approach*. London: Springer-Verlag.
- Hillier, F. S., and G. J. Lieberman. 2009. *Introduction to Operations Research*. 9th edition. New York: McGraw-Hill.
- Jacobs, P. H. M, N. A. Lang, and A. Verbraeck. 2002. "D-SOL: A Distributed Java Based Discrete Event Simulation Architecture." In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C. H. Chen, J. L. Snowdon, and J. M. Charnes, 793-800. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Pflug, G. C. 1996. *Optimization of Stochastic Models*. Dordrecht, the Netherlands: Kluwer Academic Publisher.
- Traoré, M. K., and A. Muzy. 2006. "Capturing the Dual Relationship Between Simulation Models and their Context." *Simulation Modelling Practice and Theory* 14(2):126-142.
- Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of Modeling and Simulation*. 2nd ed. San Diego: Academic Press.

AUTHOR BIOGRAPHIES

RONALD APRILIYANTO HALIM is a PhD candidate in the Transport, Logistic section at Delft University of Technology, in faculty of Technology, Policy and Management. His research is focused on the development of a global freight logistics simulation model which will be used to support decision/policy makers to deal with uncertainties in the future. Ronald holds MSc. in Systems Engineering, Policy Analysis, and Management (SEPAM) from Delft University of Technology. During his graduate study, Ronald is specializing in two areas: (TIL) transport, infrastructure, and logistics and (MSO) modeling, simulation and optimization. His research interests include evolutionary computation, simulation of freight logistics system, adaptive policy making, and simulation-based optimization. His e-mail address is R.A.Halim@tudelft.nl.

MAMADOU D. SECK received his PhD degree from the Paul Cezanne University of Marseille and his MS and M.Eng Degrees from Polytech' Marseille, France. He is currently an Assistant Professor in the Systems Engineering section in the Technology, Policy, and Management department of Delft University of Technology. His research interests include modeling and simulation formalisms, dynamic data driven simulation, human behavior representation and social simulation, and agent directed simulation. His e-mail address is M.D.Seck@tudelft.nl.