

# TimeNET Optimization Environment

## Batch simulation and heuristic optimization of SCPNs with TimeNET 4.2

Christoph Bodenstein  
System & Software Engineering, Ilmenau  
University of Technology, P.O. Box 100 565  
D-98684 Ilmenau  
Germany  
Christoph.Bodenstein@tu-ilmenau.de

Armin Zimmermann  
System & Software Engineering, Ilmenau  
University of Technology, P.O. Box 100 565  
D-98684 Ilmenau  
Germany  
Armin.Zimmermann@tu-ilmenau.de

### ABSTRACT

In this paper a novel tool for simulation-based optimization and design-space exploration of Stochastic Colored Petri nets (SCPn) is introduced. The working title of this tool is TimeNET Optimization Environment (TOE).

The targeted users of this tool are people modeling complex systems with SCPNs in TimeNET, who want to find parameter sets that are optimal under a chosen function. It allows users to create and simulate sets of SCPNs and to run different optimization algorithms based on parameter variation.

The development of this tool was motivated by the need to automate and speed up tests of heuristic optimization algorithms. A result caching mechanism is used to avoid recalculations.

### Keywords

SCPn, Simulation based optimization, Petri nets

## 1. INTRODUCTION

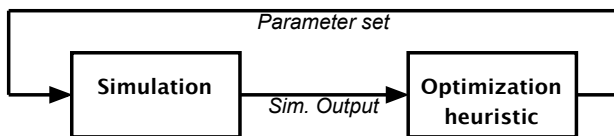


Figure 1: Common black box optimization, see [4]

Simulation-based optimization is a useful technique when the shape of the fitness function to be optimized is unknown or very complex so that gradient-based optimization algorithms cannot be applied [6, 7, 3, 8, 14]. As the modeling of real-life systems often results in complex models, simulation-based optimization has become a popular engineering design method in over last years.

Carson and Maria [4] review an overview of approaches to optimize these systems. The common part of most methods as shown in Figure 1 is a so-called black box optimization.

At the beginning of an optimization the heuristic algorithm does not know anything about the model. It only generates an input (parameter set), triggers the simulation combined with input data, and uses the generated simulation results afterward to calculate the next input until a stop condition is met.

In our work SCPNs [11] are used as a basis to model complex systems. For editing and simulating these nets we use TimeNET, a software tool developed originally at TU Berlin and now maintained at TU Ilmenau [12]. Our focus for optimization are heuristic methods, a subset of simulation-based optimization methods [4].

The idea of coupling an optimization tool and a simulation tool via defined interfaces is not new, compare e.g. [10]. Some of the features and requirements from the mentioned tool apply to our use cases and are adopted, while others were not feasible or useful for our application. The overall architecture of [10] and the generic exchange format for models and results made it impractical to reuse it with TimeNET. This led to the development of our own specialized tool.

## 2. KEY FEATURES

TimeNET is a tool for modeling and simulation of Stochastic Colored Petri nets, among other model classes and analysis algorithms. One key feature is the parametrization of nets which means that users can easily change properties of SCPNs by defining and modifying parameters. These parameters can be used in timing functions, measurements, and as place capacities.

To simulate not only one SCPN but a set while varying specific parameters can expose interesting dependencies and improve system understanding. One possibility to do several simulations is to use the TimeNET-internal scripting engine. However, this has some disadvantages regarding flexibility, debugging options, and usability. Furthermore, some features are not possible to implement with this technique (see below).

From our experiences with TimeNET and the targeted analysis of simulation-based optimization we identified some key requirements as listed below:

- Batch simulation of SCPNs while iterating several internal parameters
- Batch simulation while iterating external parameters (confidence interval, seed, maximum relative error)
- Parsing of simulation results and application of optimization heuristics
- Modular software architecture to add other heuristics and simulation types later

The first point is obvious and could have been realized with a script inside TimeNET. It allows to explore the whole expected design space step by step with a user-defined step size for discretization of continuous parameters. Afterwards the result data need to be read and plotted to get an overview of the shape of the fitness function (measurements) from the SCPN.

Further extensions such as the adaptive optimization [13] make it necessary to determine the influence of external simulation parameters like the maximum relative error or the confidence interval. Some changes were necessary inside TimeNET itself. As normal parameters are stored within the SCPN .xml file, external parameters have to be handed over to TimeNET in another way. We decided to exchange external parameters via the SCPN filename.

Since the release of the current version 4.2, there are several command-line parameters for the TimeNET GUI. One of these is the SCPN filename to load at startup. Other parameters let TimeNET automatically start a stationary simulation after the net has been loaded and to close the main program after simulation, to facilitate scripting and embedding in tool chains.

The proposed tool extension TOE uses TimeNET just as a command-line tool which is called for every simulation run. The results are read from the created log files afterwards. This is the third requirement and necessary for implementing optimization heuristics. The tool parses the result log files and stores the measured data as internal objects for later use, or exports them as a summary .csv file.

Parsing the log files and storing them as internal data structures allows the integration of optimization heuristics.

Some further requirements came up during the development. One is the ability to plot the simulation results. The user can create a script for the statistics tool R [2] and call it from within TOE to create and show simple 2D or 3D scatter-plots of chosen data from any .csv file containing simulation data.

To save time while testing the optimization heuristics, the user can load results from former batch simulations and use them as a result database instead of simulating the real SCPN again. For simple heuristic-tests this is very useful,

but for a real optimization run this database can also be used as a cache. That means whenever the optimizer creates a new set of parameters, which has already been simulated in a former batch-run, the software will choose the results from the cache. This results in a significant speedup of optimization if the used design space discretization is similar to the one used for the loaded result database. Moreover, it makes the comparison between different heuristics simpler, as the randomness of the underlying simulation is (at least partially) avoided.

A similar caching mechanism has been implemented to be used for every single optimization run.

Figure 2 shows the main window of TOE. On the left there is a table to show all possible parameters including start and end value. In combination with the step size this specifies the size and boundaries of the design space to be explored. External parameters are marked grey.

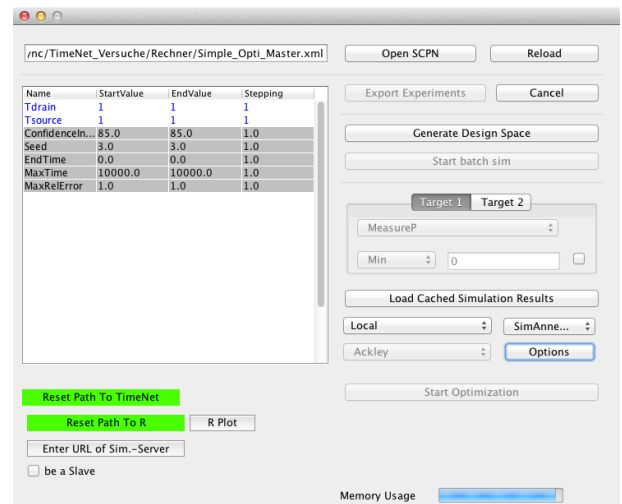


Figure 2: Main window of TOE

### 3. SYSTEM REQUIREMENTS

As the tool is implemented completely in Java, it is runnable on most standard systems which support **Java 1.6** or later. It has been successfully tested on Windows 7, OSX 10.9.4, and Ubuntu 12.2 so far.

As an obvious prerequisite, **TimeNET 4.2** [1] has to be installed to handle command line parameters. TimeNET itself needs some additional libraries on Unix-like systems and a **GNU C compiler** (or MinGW on Windows).

To plot the generated simulation results, **R** [2] needs to be installed including the **3D-plot** [5] library.

At the first program start the user should click the buttons for "Path to R" and "Path to TimeNET" and navigate to the according directories.

### 4. WORKFLOW

A simple workflow for batch simulation or optimization is shown in Figure 3. We tried to improve usability and force

the user to follow this workflow by deactivating all unnecessary buttons each time the software changes its major states. That means users cannot start an optimization while the design space is not defined or too small.

After starting the software, the last used SCPN is loaded automatically. If the program was not used before the user has to choose an SCPN file with "open SCPN". TOE will parse the corresponding .xml file and show all found parameters in the table as well as all possible measures in the tabbed pane for optimization targets.

Now the design space can be defined by setting the start/end value and the discretization value (step size). The program will automatically estimate the corresponding design space size for changed values in the parameter part. The creation of all possible parameter combinations for batch simulation can be started manually ("Generate Design Space").

Instead of defining the design space it is possible to load simulation results from a given .csv file ("Load cached simulation results"). In that case the values for start, stop, and step will be changed to match the loaded (and then cached) data. In addition to that, the option to simulate "cache-only" will be activated. The user can limit the design space by setting other values for that. If these values extend the range of the cached data, the cache-only simulation option will be disabled.

The user can then either start a batch simulation or optimization. For an optimization there is no need to generate the whole design space first, but the measure to be optimized and its target value has to be set. The predefined standard optimization algorithm is Hill Climbing. Other algorithms like Simulated Annealing or several population-based algorithms are currently being implemented.

Optimization results and some other information are printed in the log-window and (if activated) saved in the program log file. To get a quick overview of simulation and optimization results, the user can open the R-plugin, choose the appropriate .csv file as well as the data-columns to plot, and start the generation and execution of an R-script to show the results in a simple scatterplot (Figure 4 shows an example).

## 4.1 Results

By applying agile software development methodology we extended the functionality of the software over the past year. At the moment the core features are implemented and simple experiments can be run, including

- Explore the coarse discretized design space of an SCPN
- Visualize the design space in 2D and 3D scatterplots
- Test several optimization heuristics online (with real simulation) and offline (with cached result data)
- Test optimization heuristics with simple and fast benchmark functions to emulate a simulation algorithm

One typical use case at the moment is to measure the influence of simulation precision on simulation time and used

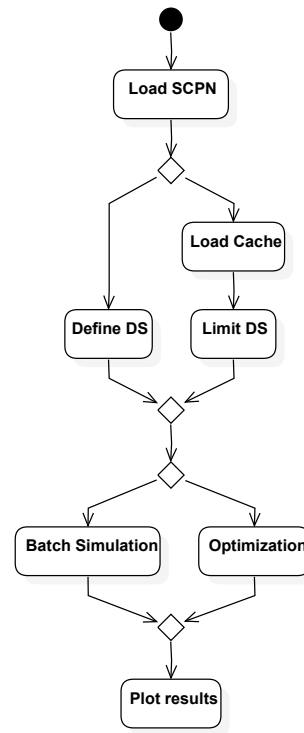


Figure 3: Typical workflow with TOE

CPU time. As an example for this, Figure 4 shows the relation between the chosen confidence interval, maximum relative error, and necessary CPU time for a stationary simulation of a simple SCPN. These results will guide the further development of the tool and lead to the integration of more sophisticated optimization heuristics [13].

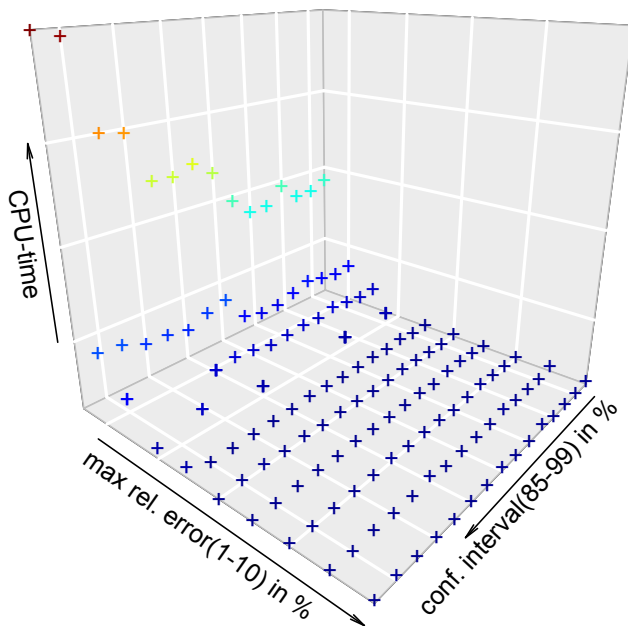
## 4.2 Restrictions and Future Work

Despite the implemented features of TOE there are several open points to be improved in the future. Optimizers and simulators have to implement a specific interface which makes it easy to integrate other optimization heuristics or other types of simulators. However, besides this interface, developers usually need to learn a lot about the internal structure before they can integrate other simulators/optimizers. A plugin-based software architecture would be very useful especially for the integration of new heuristics [9].

The external binding to TimeNET is realized via specific filename conventions and start parameters at the moment. An integration of other Petri net types such as eDSPNs would thus require some changes in TimeNET itself. On the other hand, the TOE parser for result log files would need to be adapted for other kinds of simulations and Petri nets.

Other optimization heuristics like adaptive optimization [13] are being implemented and planned to be available within the next months.

A major step forward for batch simulation and population based optimization is already visible in the interface: distributed simulation of SCPNs. TOE can run on several com-



**Figure 4: Maximum relative error and confidence interval in relation to used CPU time**

puters as a client. It will receive simulation tasks (SCPNs & simulation parameters) from a coordinating simulation server and return the results via a server to one Master instance of the tool.

Besides this some user interface improvements are planned to make the tool easier to use and to avoid unexpected behavior.

## 5. CONCLUSIONS

The paper presents an extension tool for TimeNET supporting optimization heuristics and design space exploration. The tool allows to easily test implemented algorithms based on live simulation results, cached simulation results, mixes of both, as well as benchmark function results.

The tool is written completely in Java with an extendable architecture. The current version allows basic analysis of SCPN design spaces and optimization. Other features such as distributed simulation, more heuristics and other Petri net types will be available with the next major releases.

## 6. REFERENCES

- [1] [www2.tu-ilmenau.de/downloadtimenet](http://www2.tu-ilmenau.de/downloadtimenet), September 2014.
- [2] [www.r-project.org](http://www.r-project.org), September 2014.
- [3] J. Biel, E. Macias, and M. Perez de la Parte. Simulation-based optimization for the design of discrete event systems modeled by parametric petri nets. In *Computer Modeling and Simulation (EMS), 2011 Fifth UKSim European Symposium on*, pages 150–155, 2011.
- [4] Y. Carson and A. Maria. *Simulation optimization: Methods and applications*, 1997.
- [5] [cran.r-project.org/web/packages/plot3d](http://cran.r-project.org/web/packages/plot3d), September 2014.
- [6] M. C. Fu. A tutorial overview of optimization via discrete-event simulation. In G. Cohen and J.-P. Quadrat, editors, *11th Int. Conf. on Analysis and Optimization of Systems*, volume 199 of *Lecture Notes in Control and Information Sciences*, pages 409–418, Sophia-Antipolis, 1994. Springer-Verlag.
- [7] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [8] S. Künzli. *Efficient Design Space Exploration for Embedded Systems*. Phd thesis, ETH Zurich, Apr. 2006.
- [9] R. Maschotta, S. Jäger, T. Jungebloud, and A. Zimmermann. A framework for agile development of simulation-based system design tools. In *Proc. IEEE Int. Systems Conference (SYSCON'13)*, Orlando, USA, Apr. 2013.
- [10] M. Syrjakow, E. Syrjakow, and H. Szczerbicka. Tool support for performance modeling and optimization. *International Journal of Enterprise Information Systems*, 2005.
- [11] A. Zimmermann. *Stochastic Discrete Event Systems. Modeling, Evaluation, Applications*. Springer-Verlag New York Incorporated, Nov. 2007.
- [12] A. Zimmermann. Modeling and evaluation of stochastic petri nets with timenet 4.1. In *Performance Evaluation Methodologies and Tools (VALUETOOLS), 2012 6th International Conference on*, pages 54–63, Oct 2012.
- [13] A. Zimmermann and C. Bodenstern. Towards accuracy-adaptive simulation for efficient design-space optimization. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1230–1237, oct. 2011.
- [14] A. Zimmermann, D. Rodriguez, and M. Silva. A two phase optimisation method for Petri net models of manufacturing systems. *Journal of Intelligent Manufacturing*, 12(5/6):409–420, Oct. 2001. Special issue "Global Optimization Meta-Heuristics for Industrial Systems Design and Management".