

SOFTWARE ARCHITECTURE FOR PARALLEL SIMULATION OPTIMIZATION

P. Horejsi

Department of Industrial Engineering and Management
Faculty of Mechanical Engineering
University of West Bohemia, Plzen, Czech Republic

e-mail: tucnak@cbox.cz

Manufacturing simulation issues can be solved in a variety of different COTS (Commercial off The Shelf) simulation software – for example ARENA, QUEST, SIMUL8, Siemens PlantSimulation, etc. The input and output values can be stored in databases, but the interface has to be implemented by a developer. The output values (such as manufacturing time, average time of product waiting, etc.) from one simulation run (simulator) are not the optimized values, but only the result from a particular run. The optimized values can be found after performing more simulation experiments. The number of experiments required increases with the complexity of the model and the independent input values. There is also the need to set up new input parameters correctly with consideration of optimization gradient prediction. If the model is very simple and accuracy is not important, the input settings can be made manually. For common cases this process must be automated. A software solution for the automation of the simulation optimization process in parallel logic SIMD (Single Instruction – Multiple Data Stream) was developed using Rockwell ARENA simulation software. The simulation can be speeded up rapidly with the aid of a parallel approach. This software also supports networking and also acts as the reference for further work on methodology for the optimization method and its choice of presets.

Keywords

Simulation, optimization, simulation optimization, optimization algorithm, ARENA

1. Introduction

A crucial strategic activity of an industrial company is to propose its own production systems as well as possible. These production systems and production processes have to be well controlled, innovated and improved. One of the approaches to obtaining an efficient production system is to search for optimal solutions within the simulation models of a real system. The question is how to systematically vary the input parameters of the model of the production system in such a way that reaches the best configuration of the model behaviour for a specific criterion and to do it automatically. An optimization module is usually part of a large number of simulation tools which are focused on modelling and simulating continuous and discrete systems. The problem is that only a few of these integrated optimization modules can provide the possibility of setting up the optimization algorithms relative to the objective function type. The next demand on the simulation software is to solve large-scale model experiments at short notice. One possible solution is to use parallel simulation. We concentrate on some optimization methods (which can be applied to a production system simulation model of the industrial company) and its particular software application. [Raska 2008b]

Figure 1 shows an example of a possible manufacturing simulation input. All the dependent multi-value output could be represented by one value – which is (for this case) manufacturing time τ_1 . The criteria (penalization) function value is based upon this output.

The indicated optimization approach is also known in other COTS simulation software. For instance, in COTS PlantSimulation it is possible to vary input parameters at a given range and trace defined output function, but this way of optimization is limited to one specific genetic algorithm (with networking support). Rockwell ARENA includes the relatively powerful tool OptQuest. This add-in supports only basic optimization algorithms without support of genetic algorithms, it does not support parallelization of tasks and it is not possible to compare speed and advantageousness of the applied algorithm with the ideal state. All these problems are solved by the proposed architecture (with added value: the proposed architecture is "open", enabling it to be modified at any time.). More optimization tools exist, but no others similar to the ones described, as far as author knows.

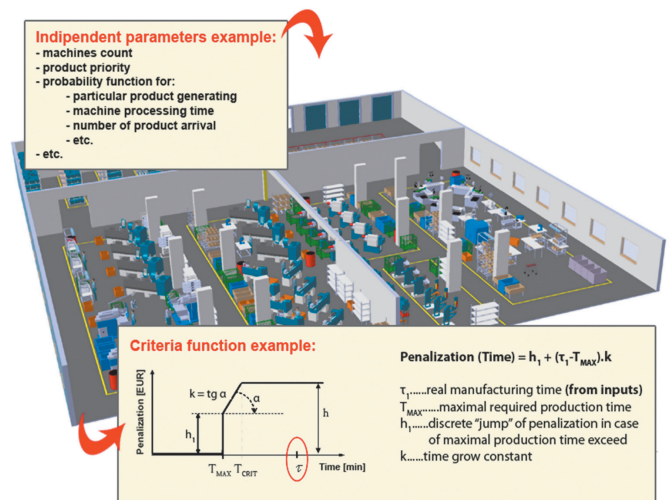


Figure 1. Input independent parameters and dependent criteria function

The author's previous work was on the project WILD (Web Integrated Logistic Development). This project involved HLA (High Level Architecture) distributed simulation of logistics and involved 7 Italian universities. An environment was developed capable of remotely executing all the involved remote simulators (federates) [Cagetti 2006].

2. Architecture

Management software called PAARE was developed, which is not only able to start the simulators but also gives feedback from the running simulators. In order to validate the proposed architecture of parallel simulation we are able to use more than 20 ARENA licenses. The model can be built at the user level with almost no need for programming skills in ARENA. So the models can be easily created and also modified. Input and output parameters are stored in a database, these parameters are modified by a model (which means only a small modification by the user of the provided sample code). Then we need to use the parallel simulation approach – we would like to start and run all the simulators in ARENA at the same time on all computers. There is also a need for a management tool in this case.

This software is able to:

- manage parallel simulation;
- configure the model parameters at client level;
- send the same model with different parameters to remote hosts – servers;
- start the ARENA model on the server(s);
- control of the remote simulation processes;
- download the result to a database;
- data-mine the database in order to get monitored parameters (for example: capacity in a particular workplace; rate of particular product in a workplace etc.);

- analyze and evaluate the monitored parameters;
- set a new simulation set and establish a new simulation cycle until the terminal condition is met;
- Find the global optimum of the tracked function;
- Organize results and generate 2D and 3D visualizations.

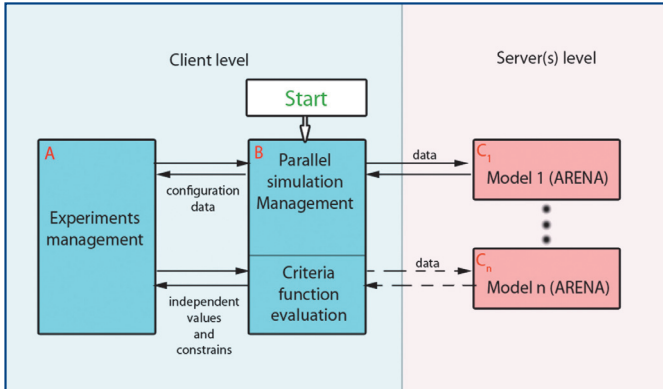


Figure 2. Simplified Concept of Automatic Management Simulation

We can see the basic “non-programming” concept of the software solution in Figure 2. The blue areas represent the main basic CLIENT module (module A and B), which deals with the model configuration and optimization configuration management. The client also has the ability to collect and process results. The results need to be gathered at one location.

The red areas represent the SERVER remote stations (C, D). These stations are able to start ARENA and run simulation models.

A – Experiment management

This block includes information about running and idle servers. It ensures that all servers (if needed) will be occupied.

B – Simulation management

All the configuration management (databases and GUI) is included in this block. This means especially Network configuration, Input databases configuration and Optimization configuration. The user can also interact with the started simulators at this level. This layer also analyses the output data.

C- Model

This block is started on remote computers. The model itself is sent from the CLIENT. Once it is started, it “lives on its own”. It can either be paused or completely revoked from the CLIENT level.

3. Software modules

For better orientation and easier implementation the software was divided into more modules, which include:

- Participating workstations management module
- Network traffic management module
- Optimization module, which includes:
 - Input and output (dependent and independent) parameters management
 - Optimization management module
 - Modules for particular optimization methods management
 - Constraints and end conditions management
- Module for export and results processing
- Visualization module

Each module represents a large scale of functions, so let us look only at the most interesting parts in next sub-chapters from the user point of view.

a. Participating workstations management

We can see the basic proposed main form GUI at Figure 3:

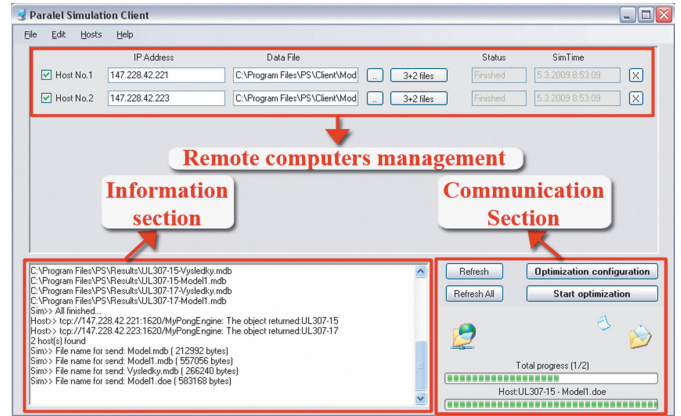


Figure 3. Main form of the application

Remote Computer Management – All the hosts participating in the simulation can be shown in this list. Their number is theoretically boundless. A host can be added, a new IP entered, a new model file added and the parameters specified for each host (downloaded and uploaded databases with the possibility of changing the input databases). The user can control each simulator after the simulation is started.

Information Section – The user can be informed about all the processes and errors in the user-feedback textbox.

Communication Section – All the model files and their parameters are deployed with the aid of this window. The user can see the remote upload and download process progress bars.

b. Optimization configuration module

The user is supposed to configure the optimization settings before the parallel optimization process (see Figure 4). The algorithm type should be set. Each of them is configured by different parameters. The following methods have been implemented so far: Hill Climbing (gradient method), Tabu Search (modified Hill Climbing), Random Search, Simulated Annealing, Simplex Method, two variants of Differential Evolution (genetic algorithm) and special method Total Search (see below). All these methods can be started in parallel logic instead of Simulated Annealing. The end condition and input and output variables can also be defined.

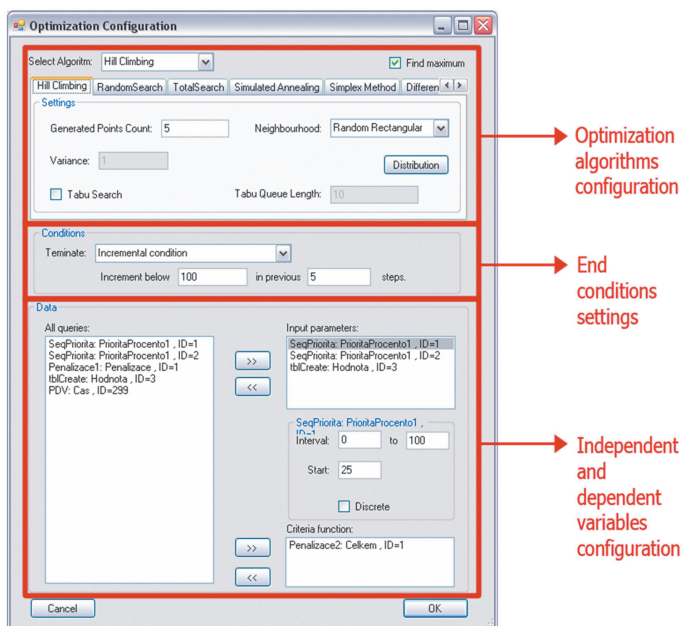


Figure 4. Optimization configuration

c. Export a visualization module

The software supports two kinds of data export: to a plain HTML report and also to Microsoft Excel. Optimization process steps are written on a table. The table format is dependent on the optimization method used. Because the output cannot be visualized on d-dimensional plain, its graph cuts are visualized instead.

When the Total Search algorithm (mapping of the criteria function behaviour) is used, we can also visualize 3D cut of the function or the whole criteria function (for 2 independent and 1 dependent variable). For an example of this see Figure 5 (direction of 3D cuts can be defined in Pivot Graph Tools).

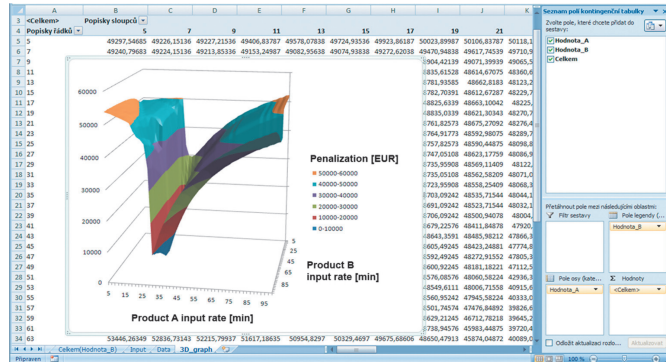


Figure 5. Total Search visualization

4. Implemented optimization methods

We will discuss the basic ideas of the implemented optimization methods in this section. The mathematical view is available in literature [Raska 2008a] and [Tvrdik 2004]. We will focus on the algorithmic view and how these methods are implemented in PAARE software. All these methods were tested for d -dimensional system. The count of independent variable is $d-1$. The goal is to find one dependent variable – value of criteria function. The value of criteria function has to be maximized (e.g. financial profit) or minimized (e.g. penalization).

To test the reliability of these methods the 'Total Search' method was also developed which is able to discover the function behaviour by interval cutting. This option is very slow and can be used for less dimensional behaviour mapping of tasks (and then used as a reference for other methods).

These end conditions are implemented in PAARE software:

- The optimization process reached the pre-entered steps count
- The maximum/minimum of criteria function is reached in the current set of optimization step
- The result is the last x steps are not changed more than declared ϵ

a. Hill Climbing

The very first implemented method is one of the "classical" methods and is also known as the Gradient Method. The origin point is set by the user (or can be randomly generated in limiting range). In the first phase the n -points are generated in the origin neighbourhood (where n is the selectable parameter). These points could be generated in the circle for each surface section or in the $(d-1)$ -dimensional rectangle. The circle generated neighbourhood is not an option for multi-scale variables. For these common cases the points are generated by the Monte Carlo method in each surface. Random numbers distribution can also be selected for each independent variable.

In the next phase all the criteria function values are counted for these generated independent points and the new origin point is set to a new maximal/minimal criteria function value. All these phases are repeated until the end condition is met.

This method is comprehensible and easy to trace. The major disadvantage is the possibility of locking in the local extreme.

b. Tabu Search

Tabu Search method is derived from the previous one. It adds the Tabu Queue principle to the concept of Hill Climbing. This FIFO queue holds up the forbidden areas which are those which have already been examined. The gradient examination could perhaps return to these areas after the chosen time which is declared by the Tabu Queue Length variable.

This method gives similar results to the previous one (with all the advantages and disadvantages) but can express faster behaviour in some cases.

c. Random Search

This method has a very simple algorithm. The n -points are generated randomly in the permitted area. From these n -points the best solution is chosen by its maximal/minimal criteria function value.

d. Simulated Annealing

The major problem with locking in the local extreme is successfully solved by the next method – Simulated Annealing. The algorithm can also accept the worse solution for a given probability (Metropolis criteria). This stochastic method can search the whole permitted area. The group of points for the next step is not derived from the previous point neighbourhood but from the stochastic failure operator.

The algorithm has the ability to unlock from the local extreme. It was quicker than Hill Climbing for non-parallel tasks. The new optimization set is generated from one point, which implies that this algorithm cannot be performed in parallel.

e. Simplex Method

For the first step of this algorithm $d+1$ non-complanar points are generated in a defined area. The BEST and WORST solution is found in simplex set S . Now we are supposed to find the centre of gravity g with the BEST solution excluded (the gravity centre is the "points average"). Now we find the new point y :

$$y = g + (g - x_{\text{WORST}}) = 2g - x_{\text{WORST}} \quad (1)$$

If the new y point does not meet condition $f(y) < f(x_{\text{WORST}})$ the reduction is applied. This reduction is applied in direction to x_{BEST} with the half distances.

$$x \leftarrow \frac{1}{2}(x + x_{\text{BEST}}), x \in S, x \neq x_{\text{BEST}} \quad (2)$$

These steps are repeated until the terminal condition is met.

f. Differential Evolution

Differential Evolution is a fast and elegant way of finding the optimum. It is frequently the default method implemented in simulation tools. This method has many more variants. In PAARE software two options are implemented: RAND and BEST.

Firstly, the basic population P is generated. The new population Q is derived from the old one by generating potential competitor y . This competitor y is generated by crossing the original vector x and the new u vector. The new population Q consists of new elements. In the new population elements from the old population could be replaced by the y vector in accordance with given probability C .

Algorithm RAND and BEST differs in generating the u vector:

$$u_{\text{RAND}} = r_1 + F(r_2 - r_3) \quad (3)$$

$$u_{\text{BEST}} = x_{\text{best}} + F(r_1 + r_2 - r_3 - r_4)$$

where r_n is the randomly and independently chosen point and F is the method input parameter.

5. Validation

The proposed software can be used as a reference for the proper algorithm and setting its parameters in the digital factory environment methodology research. The software is able to find the criteria function behaviour at a given range with the aid of the Total Search method. We can consider time-consumption, accuracy and proper parameters of other optimization methods for a given model by comparing with the results of Total Search method (a preliminary and time-consuming calculation).

One of the major benefits lies in the rapid speeding-up of the simulation process by adding computing stations. A major speeding-up could be observed for more complicated models. We need to take into account the network cost. If the real-time simulation is less than the network costs, there is no need to parallelize tasks.

More tests were performed on a network environment with low latency. For example, for 18 experiments with 22 seconds duration for each, there was a network loss of 17.5%. This loss includes model transfer to each remote computer and also the transfer of the input and output database (approximately 500kB in this case).

In Figure 6 we can see the dependency of the count of used remote computers on time needed to perform all the experiments for the case above (18 experiments per 22 minutes with consideration of network traffic loss). We can see that for the case of low time for one experiment, more computers (4 and more) have less affect because of the network losses.

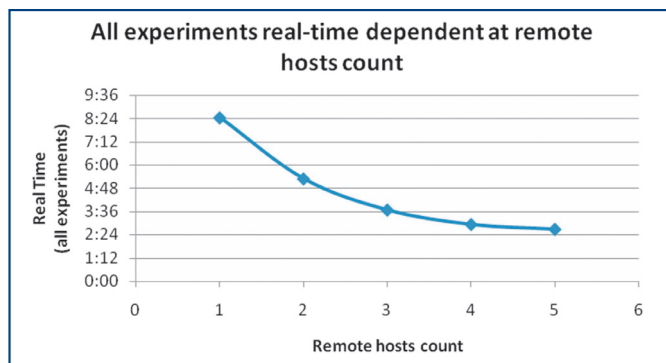


Figure 6. Graph – Dependency of real-time on remote hosts count

6. Conclusion

An elegant way has been found for optimizing manufacturing discrete event simulation tasks. The proposed solution offers a parallel networking option, but can also be used on a single computer (the user just enters the computer name, IP or 127.0.0.1 to the first remote computer). Software can be and will be modified. In particular, there will be more optimization methods added.

More students have the opportunity of using of this software. Particular experiments were performed by Ing. Michal Streda (see [Stre-

da 2009]) and Ing. Pavel Raska (see [Raska 2008a]), who will also design the reference methodology.

From the aspect of programming demands, there were 34 classes realized and 7500 rows of programming code excluding GUI proposals and resources.

References:

- [Cagetti 2006] Cagetti, M.M., Cassettari, L., Giribone, P. et al.: A Java Based Tool for Supporting High Level Architecture Federation Execution Control Over A WAN, In: *10th World Multi-Conference on Systemics, Cybernetics and Informatics/12th International Conference on Information Systems Analysis and Synthesis*, Orlando, FL, 16-19 July 2006, pp 295-300. ISBN 980-6560-72-8
- [Horejsi 2006] Horejsi, P., Candrová, K.: *Research of Methods for Optimal Management for Distributed Simulation Models of Manufacturing Systems and Their Integration to Digital Factory Concept – Digital Workshop Experiments*, Research Report, ZCU/FST/KPV, Plzen, 2006 (In Czech)
- [Kriz 2003] Kriz, P.: *Optimization Algorithms for Validation of Discrete Simulations Tasks with Aim of Machine Engineering Manufacturing*, MSc Thesis, ZCU 2003 (In Czech)
- [Raska 2008a] Raska, P.: *Analysis of Methods for Manufacturing Systems and Processes for Discrete Simulations Optimization*, Rigorous Thesis, ZČU 2008 (In Czech)
- [Raska 2008b] Raska, P., Horejsi, P.: An Architecture Proposal for Parallel Simulation Optimization. In: *MITIP 2008 Proceedings*, Prague, 12-14 November 2008, pp 50-55. ISBN 978-80-7043-738-4
- [Simon 2007] Simon, M.; Troblova, P.: Effective Design Aspects of Production System, In *Annals of DAAAM for 2007 & Proceedings of 18th international DAAAM symposium*, Zadar, Croatia, 24-27 October 2007, pp 691-692, ISBN 3-901509-58-5
- [Stefka 2005] Stefka, D.: *Alternatives to Evolution Optimization Algorithms*, MSc Thesis, CVUT 2005 (In Czech)
- [Streda 2009] Streda, M.: *Validation of Manufacturing Systems and Processes for Discrete Simulations Optimization*, MSc Thesis, ZCU 2009 (In Czech)
- [Tvrdik 2004] Tvrdik, J.: *Evolution Algorithms – study texts*, Science Faculty, University of Ostrava, Ostrava, 2004 (In Czech)
- [Ulrych 2007] Ulrych, Z., Raska, P., Horejsi, P., Candrová, K.: Basic Methodology for a Simulation Case Study Using Parallel Discrete Event Simulation. In: *MITIP 2007 Proceedings*, Florence: Florence University, 6-7 September 2007, pp 347-354. ISBN 97-88-87544-107-0

Contact:

Ing. Petr Horejsi, Ph.D.
Department of Industrial Engineering and Management
Faculty of Mechanical Engineering
University of West Bohemia
Univerzitní 8, 306 14 Plzeň, Czech Republic
tel: +420 377 63 8480, e-mail: tucnak@cbox.cz