Simulation, optimization, time reduction, optimization algorithms, application

Milan GREGOR[*], Andrej ŠTEFÁNIK[**]

OPTIMIZATION AS A TOOL FOR TIME REDUCTION IN SIMULATION PROJECTS

Abstract

This paper presents the optimization methods appropriate for discrete event simulations of manufacturing systems. The chosen optimization approach was applied in various simulation projects. Given example shows practically how can be utilize optimization in simulation. The paper presents the results of simulation and time reductions achieved through optimisation.

1. INTRODUCTION

This time period can be described as a period of innovation and cost reduction impact the companies to make many changes of production system, products and way of thinking and doing their business. Each bad decision brings increasing of cost and reducing competitiveness. Because big complexity of many systems in areas such as manufacturing, supply chain management or financial management there is convenience to use modelling and computer simulation as a powerful tool for dynamic verification of the system parameters and logical structure in the phase of its proposal and design. It can find out the bottleneck and optimize its functionality. In this way there are possibility to reduce both cost and time.

Simulation can be classified as a statistic experimental method. It is based on the simplifying the "real" system using simulation model consist of those properties describing "real" system from the point of view of the specifically simulation study. Simulation is positioned as a means to evaluate the impact of process changes and new processes in a model environment through the creation of "what-if" scenarios. Simulation is promoted to enable examination and testing of decisions prior to actually making them in the "real" environment. Usually there are many different scenarios and it is necessary to find the best of them. It is possible to check a few of them, but more exploratory process may by need in the form of simulation optimization.

[*] Prof. Ing. Milan Gregor, PhD. E-mail: milan.gregor@fstroj.utc.sk
[**] Ing. Andrej Štefánik, PhD., E-mail: andrej.stefanik@fstroj.utc.sk
Department of Industrial Engineering, University of Zilina, Faculty of Mechanical Engineering, Univerzitná 1, 010 26 Žilina, Slovak Republic

2. SIMULATION OPTIMIZATION

Simulation optimization is the process of finding the best value of some decision variables for a system where the performance is evaluated based on the output of a simulation model of the system. However the need for optimization of simulation models increases in the case finding a set of model parameters being influence the performance, there are only a handful of software offer optimization. But many times regardless of optimization module availability the user do not utilize it.

Today, there exist very powerful algorithms to guide a series of simulations to produce high quality solutions in the absence of tractable mathematical structures. Furthermore, we are now able to precisely compare different solutions in terms of quality. Many of discrete-event simulation software packages contain an optimization module that performs some sort of search for optimal values of input parameters. For example OptQuest, a leading optimization tool for commercial simulation software, employs metaheuristics such as scatter search and tabu search, and techniques such as neural networks, to provide optimization capabilities to users.

Another simulation software WITNESS utilizes its optimize module with sophisticate optimization algorithm - simulated annealing. When trying alternative combinations it can accept results that are worse than the current result. The chance of a worse result being accepted is determined by the simulated temperature which is reduced in a series of steps throughout the optimization. At the beginning of the optimization, when the temperature is higher the chance of a worse result being accepted is fairly high. As the optimization proceeds and the temperature cools it becomes less likely those worse results will be accepted. Allowing worse results to be accepted means that, the algorithm is less likely to ‘get stuck’ in a local optimum and so has more chance of finding a global optimum. It is called simulated annealing because it is analogous to annealing. This is where metal is heated and then slowly cooled, a process which relieves internal stress by encouraging stable molecular formations.

2.1. Area of simulation optimization applications

Once a simulation model has been developed to represent a system or process, we may want to find a configuration that is best, according to some performance measure, among a set of possible choices. For simple processes, finding the best configuration may be done by trial-and-error or enumeration of all possible configurations. When processes are complex, and the configuration depends on a number of strategic choices, the trial-and-error approach can be applied with only limited success. In these cases, we use an optimization tool to guide the search for the best configuration.

Selected applications may include the goal of finding the best:

- configuration of machines for production scheduling,
- integration of manufacturing, inventory, and distribution,
- layouts, links, and capacities for network design,
- investment portfolio for financial planning,
- utilization of employees for workforce planning,
- location of facilities for commercial distribution,
- operating schedule for electrical power planning,
- setting of tolerances in manufacturing design.

The optimization of simulation models deals with the situation in which the analyst would like to find which of possibly many sets of model specifications (i.e., input parameters and/or structural assumptions) lead to optimal performance. In the area of design of experiments, the input parameters and structural assumptions associated with a simulation model are called factors. The output performance measures are called responses. For instance, a simulation model of a manufacturing facility may include factors such as number of machines of each type, machine settings, layout, and the number of workers for each skill level. The responses may be cycle time, work-in-progress, and resource utilization.

In the world of optimization, the factors become decision variables, and the responses are used to model an objective function and constraints. Whereas the goal of experimental design is to find out which factors have the greatest effect on a response, optimization seeks the combination of factor levels that minimizes or maximizes a response (subject to constraints imposed on factors and/or responses). In manufacturing example, we may want to formulate an optimization model that seeks to minimize cycle time by manipulating the number of workers and machines, while restricting capital investment and operational costs as well as maintaining a minimum utilization level of all resources. A model for this optimization problem would consists of decision variables associated with labour and machines as well as a performance measure based on a cycle time obtained from running the simulation of the manufacturing facility. The constraints are formulated both with decision variables and responses (i.e., utilization of resources).

When changes are proposed to business processes in order to improve performance, the projected improvements can be simulated and optimized artificially. The sensitivity of making the changes on the ultimate objectives can be examined and quantified, reducing the risk of actual implementation. Changes may entail adding, deleting, and modifying processes, process times, resources required, schedules, work rates within processes, skill levels, and budgets. Performance objectives may include throughput, costs, inventories, cycle times, resource and capital utilization, start-up times, cash flow, and waste. In the context of business process management and improvement, simulation can be thought of as a way to understand and communicate the uncertainty related to making the changes, while optimization provides the way to manage that uncertainty.

2.2. Steps of simulation optimization

Before a model can be optimized, an objective function must be defined. This function quantifies the aim of the optimization.

During the optimization process the model will be run with different combinations of the values that are specified in the optimization variables. At the end of each run the objective function will be called to obtain the result of that run. The value of this result determines if the combination of values used was better or worse than other combinations. The optimization task is to find the combination of values that provides the best result.

In general, the objective function will be a combination of some of the current values of the optimization variables and some results of the simulation.

As a simple example; suppose a production process was being modelled and there was a choice as to the number of machines to use and staff to employ. The aim of the optimization is to produce the highest possible throughput with the lowest possible cost. The objective function could be expressed as:

value of throughput - cost of machines - cost of staff

This would then be used in the objective function.

Note that changing the relative values of the cost and value figures would result in more or less emphasis being placed on throughput as opposed to cost and so would result in a different optimum solution.

There are a number of points in a model where action statements can be executed that might change the model. The most common place for these is Initialize Actions. You must ensure that any such actions are not in conflict with the changes being made during the optimization process. Examples of actions that may cause problems are ‘SET CAPACITY’ and ‘SET QUANTITY’.

The optimization variables should be managed in the following ways:

- Add new variables.
- Change the definitions of existing variables.
- Delete one or more existing variables.
- Add constraints to existing variables (and change or delete them).

As we modify the optimization variables the total number of combinations for the values of the variables is updated. After modifying the optimization variables, we can then re-evaluate the resulting constrained combinations.

The steps of simulation optimization can by determine in 9 separate actions. These are shown in the following figure:

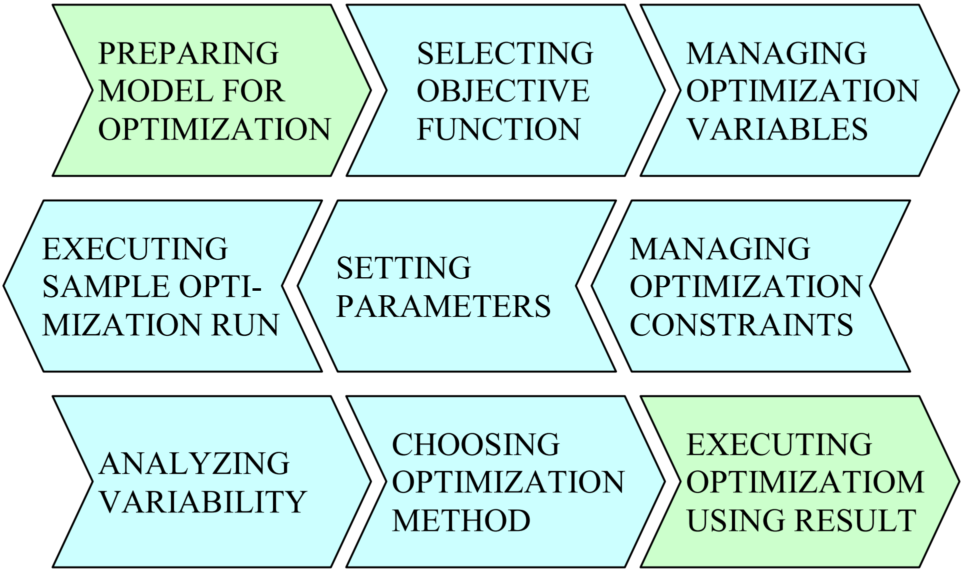


Fig. 1. Steps of simulation optimization

3. THE OPTIMIZING ALGORITHMS

There are many optimizing algorithm being used by simulation. A variety of optimization methods are provided, ranging from simply running all combinations through to the more complex and intelligent algorithms. The methods are:

All combinations, which will run all constrained combinations. If sufficient time is available, this method guarantees that the optimal result will be found. An estimation of the time to be taken by optimization can be obtained by the solving equation time of sample run multiply number of combinations.

Min/Mid/Max, which will run all combinations of min, mid and max settings of range parameters. For set parameters all values are used.

Hill Climb, which is a simple method of searching for improvements. This method iteratively generates a single neighbour, which is accepted only if it is of higher quality. The neighbourhood function selects a parameter at random; if the parameter is contiguous then it is either increased or decreased at random. Non-contiguous parameters are set to a value picked at random from their valid range.

Random solutions, which generates random combinations and can help indicate how solutions will vary, by giving a picture of the shape of the entire solution space for a particular scenario.

Adaptive Thermostatistical SA (Simulated Annealing). Although basic Hill Climb can sometimes find a local optimal solution more quickly than other techniques, this is often highly dependent on the starting values chosen by the algorithm. In contrast, Adaptive Thermostatistical SA will not get stuck in local optima and is recommended for that reason.

Six Sigma, which is based on the Simulated Annealing method. With this method you can limit the level of changes to a model for the pupose of identifying the best options for process improvement.

3.1. Simulated Annealing

Simulated annealing is an optimization paradigm based on the structural properties of physical materials that are melted down and then cooled in a controlled manner. The technique is based on the concept of local search, and designed to reduce the risk of becoming trapped in local optima. At each stage of the search, a neighbour of the current solution is generated and either accepted as the new solution or rejected. This acceptance process is initially random, but becomes increasingly dependent on solution quality as time goes on. The temperature controls the degree of randomness present within the search and is modulated by a predetermined cooling schedule.

This is the Simulated Annealing algorithm:

generate an initial solution s0 ;
select an initial temperature t0 > 0;
set current temperature t = t0;
select a cooling schedule parameter α < 1;
select a number of iterations n to spend at each temperature;
repeat
repeat
        generate a neighbour s of s0;
        δ = quality(s0) - quality(s);
        if δ < 0 then
s0 = s
        else
begin

generate random χ uniformly in the range (0,1);
if $\chi < exp(-\delta/t)$ then s0 = s;
end
until number of iterations performed at this temperature step = n;
update temperature by $t = \alpha \times t$;
until stopping condition is met

The initial temperature and cooling schedule used within Simulated Annealing is highly problem-dependent and can vary widely. The choice of cooling schedule has a considerable impact upon the quality of the resulting solution. Parameter optimization is therefore essential if high quality results are to be obtained.

The initial temperature determines the degree of randomness that is initially present in the search. Higher initial temperatures will introduce a greater degree of randomness within the search.

In the algorithm above, the cooling schedule parameter (α) controls the rate at which the temperature is reduced; large values will produce slow cooling schedules. The parameter value must always be less than one, since it must reduce the temperature, and must be greater than zero, since the temperature must not become negative, or reach zero in a single step. In practice, values of the parameter between 0.7 and 0.95 tend to be used.

The length of the temperature step also controls the cooling rate. Longer temperature steps will produce slower cooling rate if the parameter a remains fixed. The recommended number of temperature steps is approximately 25.

It is also good practice for the cooling schedule parameter and temperature step length to be set so that the final temperature is approximately 10% of the initial temperature.

3.2. Reactive thermostatistical search

Reactive thermostatistical search is a new technique that incorporates elements of tabu search into the simulated annealing process, by learning from its experience of the problem domain and modifying its search strategy accordingly. Tabu search, like simulated annealing, is based upon the local search paradigm, and in its basic form searches by iteratively moving from a single current solution to one of its neighbours, until some stopping condition is met. However, rather than introducing randomness into the search to avoid being trapped in local optima, tabu search makes use of a collection of rules to determine the nature of its search. This rule list allows tabu search to operate in a very sophisticated manner, taking into account factors such as past experience or problem domain knowledge when conducting the search.

Reactive thermostatistical search monitors the performance of each of the simulation parameters and adapts accordingly. The search gives bias towards simulation parameters which, when modified, give solutions which are accepted as replacements for the current solution.

The parameter bias is implemented within reactive thermostatistical search using an adaptive neighbourhood. Rather than selecting a simulation parameter at random to modify in the generation of a neighbour, the technique selects each parameter with probability based upon its past performance over a number of iterations χ.

3.3. Six Sigma

This is a useful algorithm for practitioners who wish to limit the amount of change from the 'current' situation in a process improvement project. The algorithm allows you to set the number of parameters that can vary from the suggested values.

For example:
In a factory that has 10 machines, it is possible to set up each of the cycle times of each machine to be a parameter, which can vary. If each parameter has the current value of the cycle time as one option and one other option of an improvement of 10% (lower cycle time) then the suggested value should be set to the current value. Then in the Six Sigma algorithm it would be possible to set a limit of, say, 4 changes. The algorithm would then try to find the best combination of four or fewer changes to the model, thus identifying the four machines that a process improvement team should concentrate on to have the best effect.

The way in which the algorithm works is based on the simulated annealing algorithm, with the additional constraint of not allowing more than χ changes. The algorithm manages this by resetting an existing changed parameter every time the limit is broken.

The additional setting on the Six Sigma is one that will run all Six Sigma combinations. This setting will override the Six Sigma algorithm and perform a full factorial experiment on the parameters whilst still enforcing the input parameter constraints and the Six Sigma constraint on the number of parameters that can change. The estimated total number of iterations when this option is selected is the total number of known combinations. In practice the actual number of iterations will be fewer than this because of parameter and Six Sigma constraints.

4. MANAGING CONSTRAINTS TO REDUCE COMPLEXITY

Constraints enable to reduce the number of possible combinations within a model in a controlled way. It can be done for two reasons. The first is to formulate additional conditions that apply to the scenario. For example, if optimization variables allow between 3 and 5 staff of one type, and between 3 and 7 of another type, you can apply a constraint condition that at any time there is to be no more than a total of 10 staff.

The second is to quantify your experience of what contributes to a good solution. For example, suppose we have a variable for the number of machines of a certain type and another variable for the number of staff who can operate such machines. If we know from experience that an operator can practically manage a maximum of three machines then we can apply a constraint that the number of machines is never greater than three times the number of operators.

Constraints are always expressed as linear equalities or inequalities combining two or more of the optimization variables. They are of the form:

$a*v1 + b*v2 + c*v3\ldots\ldots <= d$
or
$a*v1 + b*v2 + c*v3\ldots\ldots = d$
or
$a*v1 + b*v2 + c*v3\ldots\ldots >= d$

where v1, v2 & v3 are variables and a, b, c & d are constants

The first example above would have the form:

staff1 + staff2 <= 10

The second example can be expressed initially as:

machines / 3 <= operators

manipulating this to a linear inequality we get:
*machines - (operators * 3) <= 0*

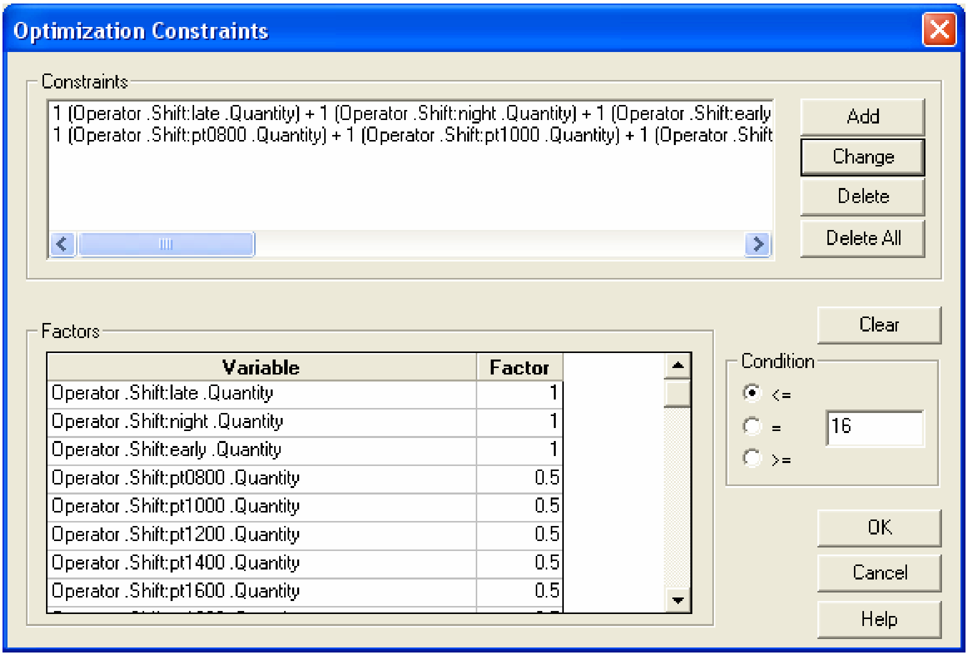


Fig. 2. Optimization constraints

5. ANALYZING VARIABILITY

One of the important things is to analyze the variability of the model. This shows how much the objective function result is affected by the changes you specified in the random number streams. A stable model will be affected less than an unstable model by the random number changes.

Among other things, this analysis can help in deciding how many runs per evaluation should be performed. An unstable model will need more runs per evaluation to achieve a reliable result.

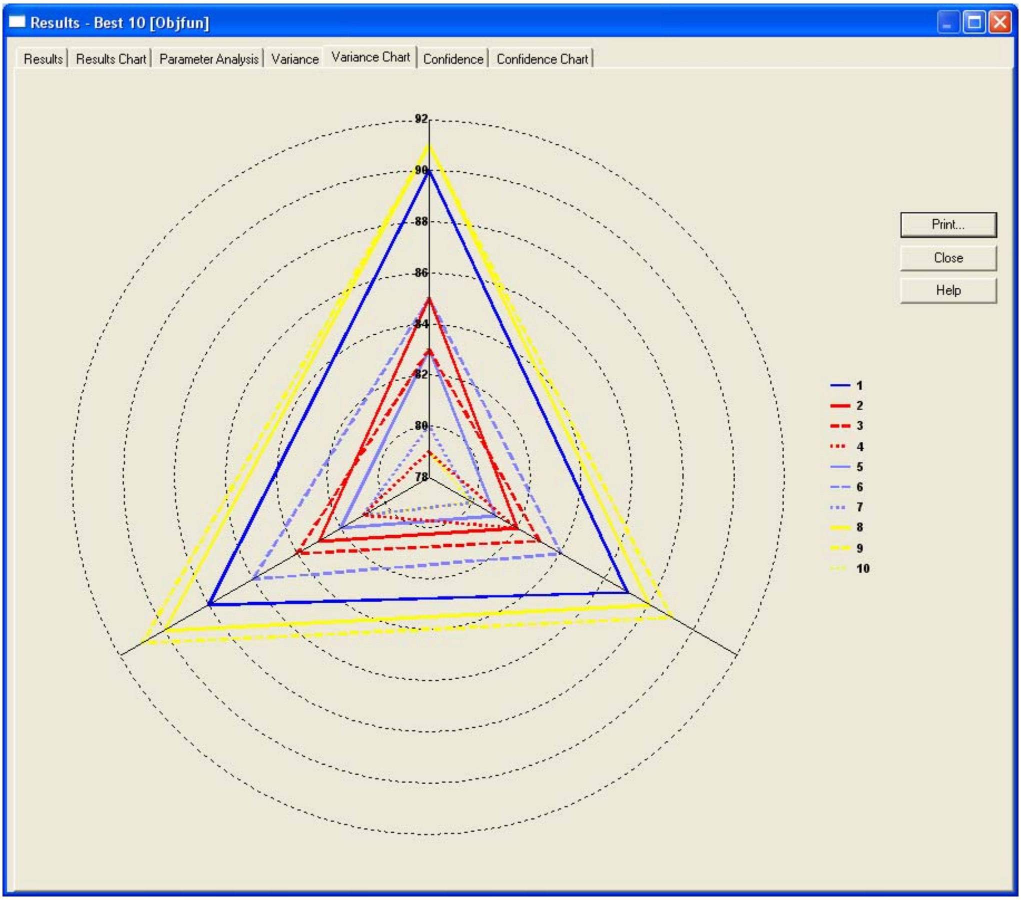


Fig. 3. Variance chart

When analyzing the model, all the variables are set to their suggested values. If a variable does not have a suggested value then its mid-point value is used.

The model is then run repeatedly. Between runs should be updated the random number streams according to the settings.

Managing the analysis involves:

1. Specifying how the results of the analysis will be displayed.
2. Initiating the analysis.
3. Viewing progress of the analysis.

6. USING THE CONFIDANCE TABLE

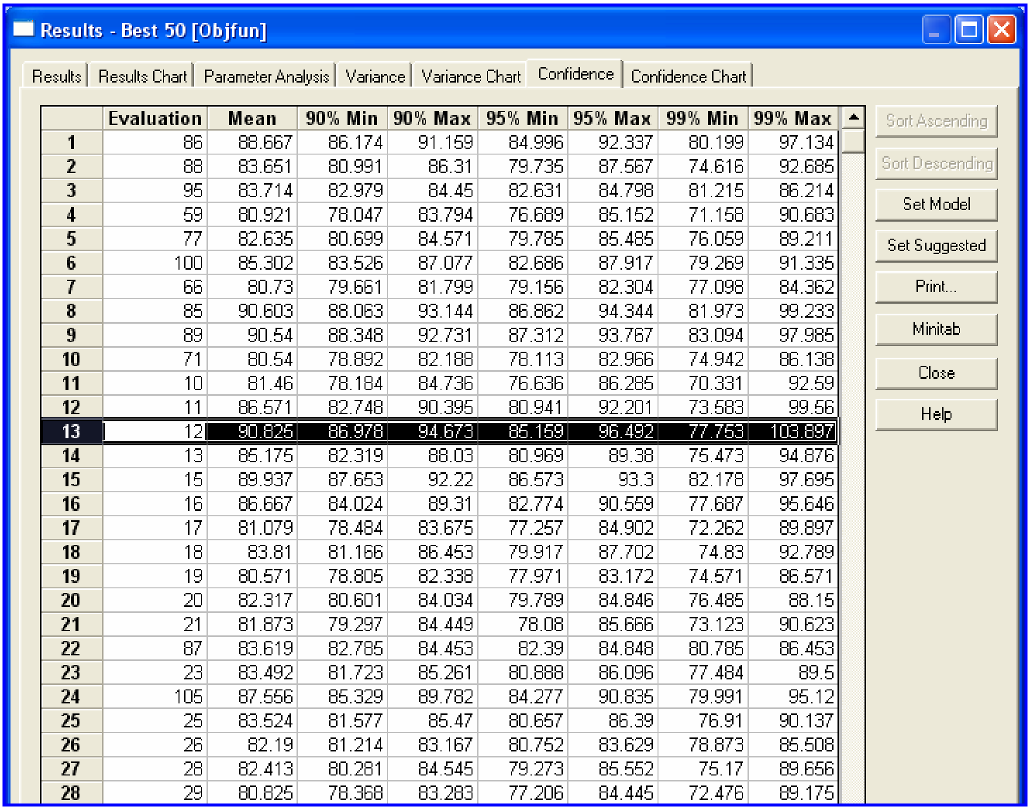
The calculated statistics give confidence limits on the mean, which has been calculated from the observations. The ‘t’ test is based on the assumption that the samples come from a Normal distribution. It is your responsibility to confirm the validity of this assumption in any particular instance.

The lower confidence limit is: $\bar{x} - \frac{(t \times s)}{\sqrt{N}}$

The upper confidence limit is: $\bar{x} + \frac{(t \times s)}{\sqrt{N}}$

Where:

- t is the Student's statistic for N-1 degrees of freedom.
- s is the standard error (the standard deviation of the distribution of estimates of the mean, generated from samples of size N, and estimated by: $s = \frac{\delta}{\sqrt{N}}$.
- N is the total number of observations.



	Evaluation	Mean	90% Min	90% Max	95% Min	95% Max	99% Min	99% Max
1	86	88.667	86.174	91.159	84.996	92.337	80.199	97.134
2	88	83.651	80.991	86.31	79.735	87.567	74.616	92.685
3	95	83.714	82.979	84.45	82.631	84.798	81.215	86.214
4	59	80.921	78.047	83.794	76.689	85.152	71.158	90.683
5	77	82.635	80.699	84.571	79.785	85.485	76.059	89.211
6	100	85.302	83.526	87.077	82.686	87.917	79.269	91.335
7	66	80.73	79.661	81.799	79.156	82.304	77.098	84.362
8	85	90.603	88.063	93.144	86.862	94.344	81.973	99.233
9	89	90.54	88.348	92.731	87.312	93.767	83.094	97.985
10	71	80.54	78.892	82.188	78.113	82.966	74.942	86.138
11	10	81.46	78.184	84.736	76.636	86.285	70.331	92.59
12	11	86.571	82.748	90.395	80.941	92.201	73.583	99.56
13	12	90.825	86.978	94.673	85.159	96.492	77.753	103.897
14	13	85.175	82.319	88.03	80.969	89.38	75.473	94.876
15	15	89.937	87.653	92.22	86.573	93.3	82.178	97.695
16	16	86.667	84.024	89.31	82.774	90.559	77.687	95.646
17	17	81.079	78.484	83.675	77.257	84.902	72.262	89.897
18	18	83.81	81.166	86.453	79.917	87.702	74.83	92.789
19	19	80.571	78.805	82.338	77.971	83.172	74.571	86.571
20	20	82.317	80.601	84.034	79.789	84.846	76.485	88.15
21	21	81.873	79.297	84.449	78.08	85.666	73.123	90.623
22	87	83.619	82.785	84.453	82.39	84.848	80.785	86.453
23	23	83.492	81.723	85.261	80.888	86.096	77.484	89.5
24	105	87.556	85.329	89.782	84.277	90.835	79.991	95.12
25	25	83.524	81.577	85.47	80.657	86.39	76.91	90.137
26	26	82.19	81.214	83.167	80.752	83.629	78.873	85.508
27	28	82.413	80.281	84.545	79.273	85.552	75.17	89.656
28	29	80.625	78.368	83.283	77.206	84.445	72.476	89.175

Fig. 4. Confidence intervals

For example, if we choose the 90% minimum column, and do ascending sort, the evaluations will be ordered starting with the lowest numerical result. If we have defined the best result as minimum, the topmost result will be the best.

7. OPTIMIZING THE STAFF IN A CALL CENTRE

The following example shows a typical application of a call centre. The calls arrive at varying rate through the day and each call can take a variable amount of time to deal with. We are the development manager for this centre and our job is fulfil performance standards with minimum staff cost.

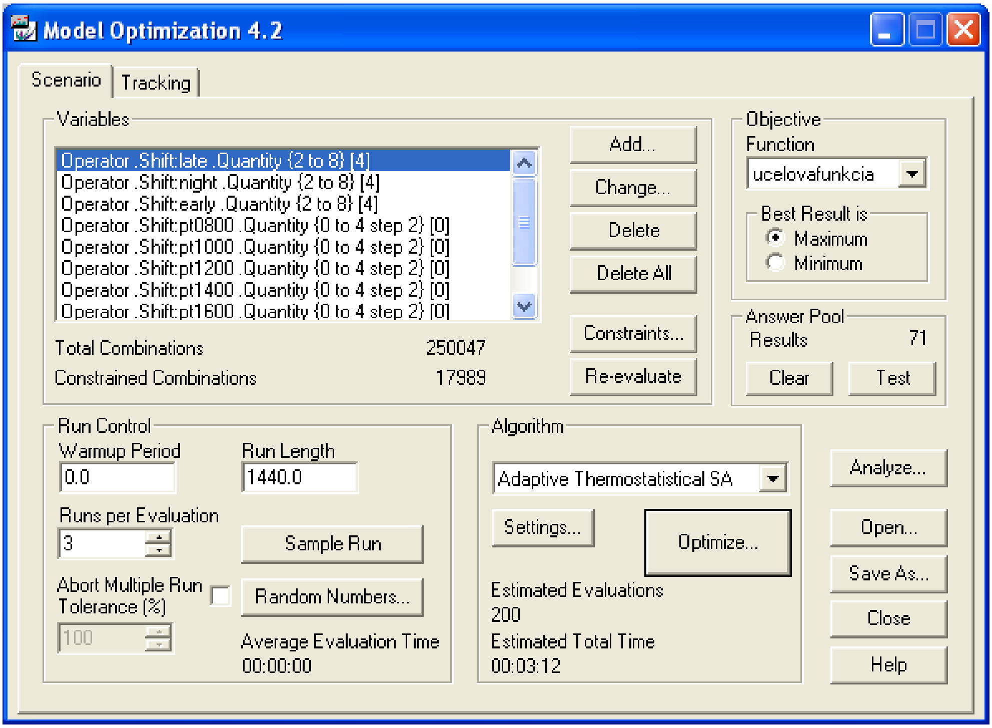


Fig.5. Window of WITNESS model optimization

Performance is measured as the percentage of all calls answered within the maximum waiting times with ideally zero abandoned calls. We are able to affect the performance by the number of staff that you have available throughout the day. This becomes a little more complicated by the fact that we may have part time staff (who work for 4 hour shifts) as well as full time staff working standard 8 hour shifts. The Call centre operates 24 hours every day. Two part time staff cost the same as one full time staff. The only constraints are that we may not have more than a total of 16 operators or the equivalent when using part time staff and no more than a total of 8 part time staff defined.

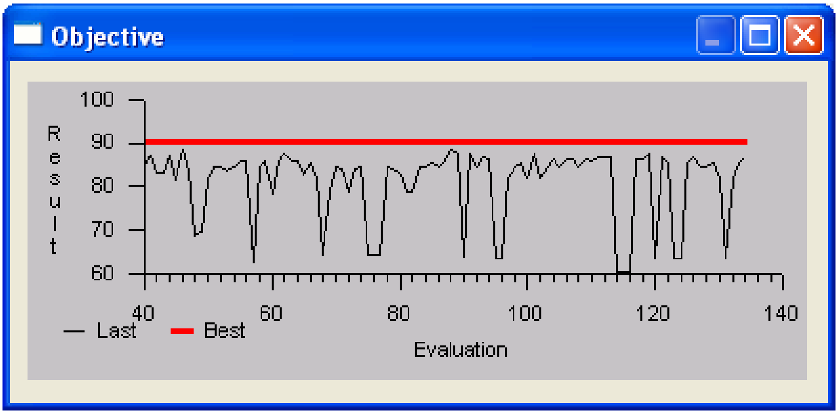


Fig. 6. Value of objective function through optimization

The result of optimization is shown in the follow table:

Results - Best 50 [Objfun]

Results | Results Chart | Parameter Analysis | Variance | Variance Chart | Confidence | Confidence Chart

Sort Ascending | Sort Descending | Set Model | Set Suggested | Print... | Minitab | Close | Help

	Evaluation	Objfun	Operator .Shift:late .Quantity	Operator .Shift:night	Operator .Shift:early	Operator .Shift:pt0800	Operator .Shift:pt1000	Operator .Shift:pt1200	Operator .Shift:pt1400	Operator .Shift:pt1600	Operator .Shift:pt1800
1	12	90.825	4	4	4	2	2	0	0	2	2
2	85	90.603	5	4	3	2	2	0	0	4	0
3	89	90.54	6	3	3	2	2	0	0	4	0
4	15	89.937	4	4	4	2	2	0	2	0	2
5	104	88.984	6	3	3	2	2	0	0	2	0
6	86	88.667	5	3	3	2	2	0	0	4	0
7	84	87.905	5	4	3	2	2	0	0	2	0
8	105	87.556	6	3	3	2	2	0	2	0	0
9	35	86.921	4	4	3	2	2	0	0	2	2
10	16	86.667	4	4	3	2	2	0	2	0	2
11	11	86.571	4	4	4	2	2	0	0	2	0
12	43	86.254	5	5	2	2	2	2	0	2	0
13	40	86.063	5	4	2	2	2	2	0	2	0
14	90	85.778	6	2	3	2	2	0	0	4	0
15	38	85.746	5	4	2	2	2	0	0	2	2
16	91	85.651	6	3	3	2	0	0	0	4	0
17	100	85.302	6	3	2	2	2	0	0	4	0
18	97	85.302	6	3	3	2	2	2	0	0	0
19	13	85.175	4	4	4	2	2	0	0	0	2
20	64	84.095	5	5	3	2	2	2	0	0	0
21	45	83.873	5	5	2	2	2	2	2	0	0
22	18	83.81	4	4	3	0	2	0	2	0	4
23	95	83.714	6	3	3	0	2	0	0	4	0
24	88	83.651	5	3	3	2	2	0	2	0	0
25	87	83.619	5	2	3	2	2	0	0	4	0
26	39	83.587	5	4	2	2	2	0	0	2	0
27	25	83.524	4	4	3	2	0	2	0	0	4
28	23	83.492	4	4	3	2	2	0	0	0	4
29	67	83.143	5	4	3	2	2	2	0	0	0
30	41	83.111	5	4	2	2	2	2	2	0	0
31	33	82.889	4	4	2	2	2	0	0	2	2
32	77	82.635	5	5	3	2	2	0	0	0	0
33	28	82.413	4	4	3	2	2	0	0	2	0
34	20	82.317	4	3	3	2	0	0	2	0	4
35	48	82.222	5	6	2	2	2	0	2	0	0
36	26	82.19	4	4	3	2	0	2	0	2	0
37	46	82.063	5	5	2	2	2	0	2	0	0

Fig. 7. Table of result of simulation optimization

In the case of 3 shifts – early, late and night being occupied from 2 to 8 fulltime operators and possibility to use from 0 to 4 part-time operators from 8 to 18 hours in step of 2 hours. So it can be assumed that the problem has 9 variables and 2 constraints. The number of overall scenarios is 250047. Being employed constraints no more than a total of 16 operators and no more than total of 8 part time operators, the number of scenarios decreases to 17898. If we want a sample size of, say, at least 30 runs per trial solutions in order to obtain the desired level of precision, than each experiment would take about 1.5 minutes. This mean that the complete enumeration of all possible solution take approximately 26847 minutes, or about 447 hours, or about 19 days. It is still too much time and variances to check each one.

Using WITNESS optimize module with Adoptive Thermostatistical Simulation Annealing algorithm and maximum of 100 movements without improvement was done 113 scenarios for the time not over the 3 hours.

8. CONCLUSION

Although most commercial simulation software products now have an embedded optimization tool, until recently, the simulation community had not embraced optimization. But the most real systems are too complex to be analyzed by trial-and-error method. In these cases is the most appropriate to use such tool to obtain the goal of simulation study. This optimization tools, along with their simulation software hosts, do not require a high level of technical sophistication from the user. This fact can support of utilizing simulation optimization in real practice.

References

[1] LAW A.M., KELTON W.D.: *Simulation Modelling and Analysis*, Second Edition, McGraw-Hill 1991, 759 s. ISBN 0-07-100803-9.

[2] APRIL J., BETTER M., KELLY J.P.: *Enhancing Business Process Management With Simulation Optimization.* In BP Trends Newsletter, White Paper & Technical Briefs, 2005, p. 1 – 11.

[3] ŠTEFÁNIK A., GRZNÁR P.: *Quick and Appropriate Changes with Support of Modelling and Computer Simulation*. In: Transcom 2005, ISBN: 80-8070-414-7, Žilina 2005, p.173-176.

[4] ŠTEFÁNIK A., GREGOR M.: *Rapid Development of the New Production System with the Support of Computer Simulation,* In. Rapid Production 2004 Innovation – Knowledge – Industry, Wroclaw, Poland 2004, p. 245-250.

[5] GREGOR M., KOŠTURIAK J.: *Schlanke Produktion braucht einfache und schnelle Simulationswerkzeuge.* Technica 1993 (in German), Nr. 25/26, S. 10-13.

[6] GREGOR M., KOŠTURIAK J., HROMADA J., BALCERČÍK V.: *How the Simulation Helps to Improve Manufacturing Systems.* In proceedings of the Second World Congress on Intelligent Manufacturing Processes and Systems. Budapest 1997. pp. 110 - 115.