TITLE OF PAPER: SIMULATION OPTIMIZATION IN MANUFACTURING SYSTEMS

RASKA, P[avel] & ULRYCH, Z[denek]

Abstract: The paper deals with the use of heuristic methods suitable for simulation optimization. These methods - Random Search, Hill Climbing, Tabu Search, Local Search, Downhill Simplex, Simulated Annealing, Differential Evolution and Evolution Strategy - automatically adapt discrete event simulation models input parameters and four analytic functions. Simulation models reflect real systems of industrial companies. Developed models are focused on logistics and production scheduling problems. Discrete event simulation models were built by Arena simulation software. Developed simulation optimization application is able to search for global optimum of the objective function by seven modified optimization methods.

Keywords: simulation optimization, discrete event simulation, global optimization, heuristic methods

1. INTRODUCTION

The crucial strategic activity of each industrial company is to design its own production systems as well as possible. The company goal is to control (innovate, improve etc.) almost every process which can be managed to obtain an efficient production system. The question is how to set up this system? Discrete event simulation with simulation optimization can be used in this case. Our department focuses on modelling and optimizing processes of industrial production systems. We have solved many projects but some of them had to be solved with difficulty without the use of a special simulation optimization tool e.g. [1] because of the large complexity of the discrete event simulation model.

A simulation optimizer automatically varies the input parameters of the discrete event simulation model of the real production system in such a way that achieves the best configuration of the simulation model behaviour for a specific criterion with respect to the defined model constraints [1]. We can say that the optimization algorithm systematically and effectively searches for optimal solutions within the simulation models of a real system. Most of today's simulation tools focus on modelling and simulating continuous and discrete systems (Arena, Witness, PlantSimulation etc.) using their own simulation optimizer. The problem is that only a few of these black-box integrated optimization modules can provide the possibility of setting up the optimization algorithms relative to the objective function type.

There is a wide variety of optimization algorithms. The question is: "Why are there so many optimization algorithms?" A simple answer could be: "Because there are many different kinds of optimization problems, and one of these methods could be much more effective than

others." Many present researchers modify optimization algorithms and develop their simulation optimization applications only for a concrete discrete event simulation problem.

For practical reasons we have developed a simulation optimizer which contains seven different optimization methods which are normally applied to analytic functions, numerical optimization and simulation optimization.

The selected methods had to be modified in such a way that they are applicable for discrete event simulation optimization purposes. We have tested and compared modified global optimization algorithms for the discrete event simulation case study and various analytic functions. The implemented methods use the evolution principle. This means that the algorithm generates more than one possible solution to avoid getting stuck on a local optimum.

The testing was performed to map the global optimization algorithm behaviour and its global optimum search efficiency, because there are many common problems of global optimization which affect finding the global optimum.

2. GLOBAL OPTIMIZATION PROBLEMS

The basic common problem of global optimization methods is to quickly find the optimum. In the case of simulation optimization, the aim is to find the global extreme (global optimum) of the objective function (respecting the constraints). It can be formulated as follows:

$$\tilde{\mathbf{X}} = \operatorname{argmin}_{\mathbf{X} \in \tilde{\mathbf{X}}} F(\mathbf{X}) = \{\tilde{\mathbf{X}} \in \tilde{\mathbf{X}} : F(\tilde{\mathbf{X}}) \leq F(\mathbf{X}) \forall \mathbf{X} \in \tilde{\mathbf{X}}\} \quad (1)$$

Where:

- $\tilde{\mathbf{X}}$... Global minimum of the objective function
- $F(\mathbf{X})$... Objective function value of candidate solution – the range includes real numbers, i. e. $F(\mathbf{X}) \subseteq \mathbb{R}$
- $\tilde{\mathbf{X}}$... Search space

Let's list some basic problems of the methods used in global optimization [2]:

1. The whole search space cannot be examined (testing all possible solutions) because of large demands on computer memory, or time spent on examining the space – nondeterministic polynomial problems.

2. Premature convergence – the global optimization process can converge prematurely to a local optimum because there is no opportunity to examine other areas of a space of possible solutions (currently, only a particular area is examined). Another area of search space exists (which is not currently known) that contains a better solution than the currently known solution.
3. The landscape of objective function – multimodal objective function – premature convergence, objective function smoothness landscape etc.
4. Multi-objective optimization.
5. Identification of a suitable method for handling the constraints.
6. Specification of appropriate termination criteria.
7. Setting the parameters of the optimization method, etc.

3. SIMULATION OPTIMIZATION APPLICATION

Our simulation optimization application Fig. 1 addresses the problems listed above:

1. Selection of optimization methods (with the knowledge of the objective function, the choice of a suitable optimization method can significantly influence the result).
2. Setting the optimization parameters setting of the optimization method.
3. Finding suitable settings for the parameters of the optimization methods – it is possible to set the number of simulation experiments with the same algorithm parameters setting (user can set lower and upper boundary of the selected parameter and the step).
4. Evaluation of the behavior of the optimization method.
5. Implementation of our own modified methods with Visual Basic programming language.
6. Visualization of current results (objective function values).
7. Management of simulation experiments.
8. Allowing the creation of a knowledge base of a simulation model (user can open a saved knowledge base).
9. Visualization of simulation experiments in 3D graph of objective function.
10. Saving simulation experimental data to database.
11. Saving simulation experiments (objective function values chart) to image.

3.1 Implemented Methods

The simulation optimizer contains seven different global optimization algorithms – Random Search, Hill Climbing, Tabu Search, Local Search, Downhill

Simplex, Simulated Annealing [3], Differential Evolution and Evolution Strategy. Previously mentioned methods are components of metaheuristic algorithms [4]. The selected methods had to be modified in such a way that they are applicable for discrete event simulation. Methods use the evolution principle – algorithms create a whole generation of individuals instead of one individual. Using this candidate solutions principle leads to keeping diversity of individuals in the initial stage of the algorithm search. We can say that the implemented algorithms use two basic diversity approaches. The first approach is to generate other new solutions which have not been investigated before - exploration.

Since computers have only limited memory, already evaluated solution candidates usually have to be discarded in order to accommodate new ones. Exploration is a metaphor for the procedure which allows search operations to find new and maybe better solution structures.

Exploitation, on the other hand, is the process of improving and combining the traits of the currently known solutions, as done by the crossover operator in evolutionary algorithms, for instance.

Exploitation operations often incorporate small changes into already tested individuals leading to new, very similar solution candidates or try to merge building blocks of different, promising individuals. They usually have the disadvantage that other, possibly better, solutions located in distant areas of the problem space will not be discovered. [5]

Fig. 1. Simulation Optimization Application IDE

4. SIMULATION MODELS

The testing of methods of global optimum searching was applied to discrete event simulation models which reflect the production systems of industrial companies. All discrete event simulation models were built in Arena simulation software. The first model represents an assembly line. Products are conveyed by conveyor belt. The assembly line consists of eleven assembly workplaces – six of them have their own machine operator. The other six workplaces are automated. A specific scrap rate is defined for each workplace. At the end of the production line is a sorting process for defective products.

The second simulation model represents a production line which consists of eight workstations. Each workstation contains a different number of machines. Each product has a specific sequence of manufacturing processes and machining time. The product is penalized if the product exceeds the specified production time. A penalty also occurs if the production time value is smaller than the specified constant. This rule is defined because of premature production which leads to increasing storage costs – the JIT product [6].

The third discrete event simulation model represents the production of different types of car lights in a whole production system. The complex simulation model describes many processes; for example, logistics in three warehouses, production lines, 28 assembly lines, painting, etc.

We tested optimization algorithms on four standard test analytic functions – De Jong's, Rosenbrock's, Michalewicz's and Ackley's function [7] - Fig. 2.

Fig. 2. Ackley's function used for optimization algorithms testing

5. OBJECTIVE FUNCTIONS OF DISCRETE EVENT SIMULATION MODELS

We specified different objective functions considering the simulated system.

In the first case the objective function of the assembly line simulation model reflects the penalty which is affected by the number of defective products and the palettes in system. The input simulation model parameters – decision variables – are the numbers of fixtures in the system and the number of fixtures when the operator has to move from the first workplace to the eleventh workplace to assemble waiting parts on the conveyor belt.

In the second simulation model the objective function is affected by the total time spent by the product in the manufacturing system. Decision variables of the production line simulation model are the arrival times of each product in the system. The objective function landscape of this model is shown in Fig. 3

The third objective function is focused on the logistics in the complex manufacturing system. The objective function is affected by the sum of the average utilization of all assembly lines, average transport utilization. Decision variables are the number of forklifts responsible for: transport of small parts from the warehouse to the production lines and assembly lines, transport of large parts from the warehouse to the

assembly lines and the transport of the final product from the assembly lines to the warehouse.

Fig. 3. Objective function landscape of the second discrete event simulation model

6. EVALUATION

Simulation experiments results are saved to excel files during simulation experiments. Excel was selected because of its wide usage. At the end of the simulation experiments the data are stored in the same excel file. The excel file contains information about:

1. Setting of algorithm parameters setting.
2. The specified number of the simulation experiments in one series.
3. The found objective function values.
4. The number of experiments required to find the global or local optimum.
5. Specified constraints.
6. Visualization of Boxplot charts - displayed for each setting (series of simulation experiments) Fig. 4:
 - a. Best objective function value.
 - b. Range of provided function objective values during the simulation experiments.
 - c. Number of experiments required to find global or local optimum.

Due to the large volume of data we have to propose five different types of fitness which evaluate each setting of simulation optimization parameters:

1. Optimization method success – finding known VTR – (value to reach).
2. Distance between found local optimum and global optimum or VTR.
3. Quartile analysis – range of provided simulation results.
4. Number of experiments required to find global or local optimum.
5. Algorithm convergence.

Fig. 4. Example of results from simulation optimization experiments provided by Evolution Strategy – Ackley's function

Fig. 5. Average method success – simulation optimization results of analytic functions

Fig. 6. Average method success – simulation optimization results of discrete event simulation models

7. CONCLUSION

The goal of our research is to develop a simulation optimization application which automatically varies the input parameters of the discrete event simulation model of the real production system in an industrial company. This research is focused on modifying heuristic methods in such a way that achieves the best configuration of the simulation model behaviour for a specific criterion with respect to the defined model constraints.

It is obvious that the success of heuristic optimization methods depends on the fitness of the objective landscape. The hardest search for the global optimum

was for the second objective simulation model. The objective fitness landscape is very rough and it contains many little peaks. If we compare the success of the optimization methods applied to the discrete event simulation models compared to analytic models we can see that the optimization methods are less successful. Some of the selected methods are actually slightly successful e.g. the Downhill Simplex Method and Differential Evolution. The most successful is Evolution Strategy and the second place is occupied by Simulated Annealing. A suitable method also could be Tabu Search, but its success is not as good as the previous two methods. Gradient methods prematurely converge. This problem can be solved by random restarts method. We would like to propose some optimization method modifications to increase their efficiency. The next step could be automatic switching of methods during the optimization with the support of artificial intelligence and using parallelization.

8. ACKNOWLEDGEMENTS

This paper was created with the subsidy of the project SGS-2012-063 named “Integrated production system design as a metaproduct with use of a multidisciplinary approach and virtual reality” carried out with the support of the Internal Grant Agency of University of West Bohemia.

9. REFERENCES

- [1] Votava, V.; Ulrych, Z.; Edl, M.; Korecký, M.; Trkovský, V. (2008). Analysis and Optimization of Complex Small-lot Production in new Manufacturing Facilities Based on Discrete Simulation, *Proceedings of 20th European Modeling & Simulation Symposium EMSS 2008*, Genova, ISBN 978-88-903724-0-7, pp. 198-203, Diptem Università di Genova, Genova
- [2] Technologies, Rockwell Automation., *OptQuest for Arena - user's guide*. Orlando, USA : Rockwell Software, 2006.
- [3] Mařík, V. ; Štěpánková, O. ; Lažanský, J. (2001). *Artificial intelligence Vol. III.* (in Czech language Umělá inteligence (3)), Akademie věd České republiky - Academia Praha, ISBN 80-200-0472-6, Prague
- [4] Kwang, Y. L. & Mohamed, A.E.S. (2008). *Modern Heuristic Optimization Techniques - Theory and Applications To Power Systems*, IEEE Press, ISBN 978-0471-45711-4, New Jersey
- [5] Michalewicz, Z. & Fogel, D. B. (2004). *How to Solve It: Modern Heuristics*, Springer, ISBN 978-3-540-22494-5, Berlin
- [6] Weise, T. (2009). *Global Optimization Algorithms - Theory and Application 2nd Edition*, Available from: <http://www.it-weise.de/projects/book.pdf> Accessed: 2012-02-02
- [7] Votava, V. & Ulrych, Z. (2006). Selected methods for optimization needs VVSDV, Internal Grant (in Czech language Vybrané optimalizační metody pro potřeby VVSDV, Interní grant), University of West Bohemia, Pilsen
- [8] Tvrdík, J. (2004). Evolutionary algorithms - Textbooks (in Czech language Evoluční algoritmy - učební texty), Available from: http://prf.osu.cz/doktorske_studium/dokumenty/Evolutionary_Algorithms.pdf Accessed: 2008-02-06
- [9] Weise, T. (2008). Simple Interface for Global Optimization Algorithm – SIGOA, Available from: www.sigoa.org Accessed: 2008-12-10
- [10] Hynek, J. (2008). *Genetic algorithms and genetic programming* (in Czech language Genetické algoritmy a genetické programování), Grada, ISBN 978-80-247-2695-3, Prague