Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 100 (2015) 807 – 815

Procedia Engineering

www.elsevier.com/locate/procedia

25th DAAAM International Symposium on Intelligent Manufacturing and Automation, DAAAM 2014

Comparison of Modified Downhill Simplex and Differential Evolution with other Selected Optimization Methods used for Discrete Event Simulation Models

Pavel Raska*, Zdenek Ulrych

Department of Industrial Engineering - Faculty of Mechanical Engineering, University of West Bohemia, Univerzitni 22, Pilsen 306 14, Czech Republic

Abstract

The paper deals with testing and evaluation of selected heuristic optimization methods - Random Search, Downhill Simplex, Hill Climbing, Tabu Search, Local Search, Simulated Annealing, Evolution Strategy and Differential Evolution. We modified basic methods in such a way that they are applicable for discrete event simulation optimization purposes. The paper is mainly focused on testing Downhill Simplex and Differential Evolution because these methods achieved below-average performances in the initial testing of finding the global optimum. We modified these methods and we compared the modified and previous basic versions of these methods. We proposed different evaluation criteria (criteria express the success in different ways). These criteria use box plot characteristics calculated from the repeated optimization experiments. We have also tested different settings of these optimization methods to analyse their behaviour considering the setup of the optimization method parameters.

Peer-review under responsibility of DAAAM International Vienna

Keywords: Downhill Simplex, Differential Evolution; simulation optimization; discrete event simulation models; heuristic methods

1. Introduction

There are many optimization methods that tell us which factors have to be changed to ensure the effective behaviour of a production system in an industrial company. Most industrial companies have very large, complex and intricate production systems. The system can be replaced by a simulation model. This gives us the opportunity to

* Corresponding author. Tel.: +4-203-776-38415; fax: +4-203-776-38402.
E-mail address: praska@kpv.zcu.cz

Peer-review under responsibility of DAAAM International Vienna
doi:10.1016/j.proeng.2015.01.435

test different variations of behaviour under different specified conditions e.g. [1]. Simulation can provide us with often unexpected results that can detect previously overlooked errors in the system e.g. [2]. Another related area is the area of simulation optimization. Problems in the optimization of production processes are often nondeterministic polynomial (NP hard) problems, where it is not possible to calculate all the possible variations to find the optimum. The goal of the simulation optimization is to test various solutions to find a suitable setting of the simulated system under specified conditions (feasibility of solution). Many of today's simulation software used for modelling company production processes (e.g. Arena [3], Witness [4], PlantSimulation [5], etc.) use their own integrated simulation optimizers. [6]

Unfortunately, these optimizers are "Black boxes" trying to combine different methods in order to find appropriate solutions (in the best case "global optimum", i.e. the best solution was found and there is no better solution) in most cases and therefore the optimization method used for the solution cannot be identified; user cannot set the parameters of the optimization method (optimization methods parameters are managed by an optimizer internal algorithm); the user cannot usually implement his own methods; the user cannot specify a termination criteria for the optimization run; creating a simulation model in the supplied software – user cannot use the simulation model created in another simulation software; there is no possibility to save simulation experimental data to a database (or to create a knowledge database), etc.

We have developed our own simulation optimization application which addresses the problems listed above. It is also a reflection on the research needs of both academia and the practical sphere. The user can perform a detailed analysis of the simulation results. The simulation optimizer can optimize the simulation models built in Arena and PlantSimulation simulation software. The user can also specify an objective function describing the behaviour of the modelled reality using the simulation optimizer's graphical user interface.

2. Discrete event simulation models and testing functions

We tested the optimization methods' abilities to search for the global optimum for three discrete event Arena simulation models. These models reflect real production systems in Czech industrial companies. Each model has a specific objective function considering the simulated system and the simulation goal. The entire search space of each simulation model was mapped to find the global optimum of the objective function. [7]

- The Manufacturing System and Logistics model - this 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. The objective function is affected by the sum of the average utilization of all assembly lines and average transport utilization. The objective function is maximized. Controls 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.
- The Penalty model - this 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 times. 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 premature production leads to increasing storage costs – the JIT product). The objective function is affected by the total time spent by the product in the manufacturing system. The objective function is minimized. Controls of the production line simulation model are the arrival times of each product in the system.
- The Assembly Line model - this model represents an assembly line. Products are conveyed by conveyor belt. The assembly line consists of eleven assembly workplaces. Six of these workplaces have their own machine operator. The rest of the 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 objective function reflects the penalty which is affected by the number of defective products and the palettes in the system. The objective function is maximized. The input simulation model parameters (controls) are the number 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.

We tested the implemented optimization methods on four standard testing functions – De Jong´s, Rosenbrock´s, Michalewicz´s and Ackley´s functions. All visualizations of the objective function of the simulation model can be found in reference [8]. All testing functions were minimized.

3. Implemented optimization methods

An overview of the optimization methods used can be found in the following literature ([9], [10], [11], [12], [13], [14]).

We have identified some optimization methods which are commonly used by simulation optimizers which are integrated in most recent simulation software. Selected methods are: Random Search, Downhill Simplex [15], Hill Climbing, Tabu Search, Local Search, Simulated Annealing [16], [17]Evolution Strategy [18], [19] and Differential Evolution [20], [21]. Previous testing of optimization methods confirms that generating one solution leads to premature convergence in most cases (depending on objective function type). Different variants of selected optimization methods obtained from a literature review were united into the algorithm. The user can combine different variants of optimization methods by setting the optimization method parameters. We specified the following termination criteria – value to reach, exceeding the computational time, achieving a specified total number of iterations and no improvement. We have slightly modified the optimization methods regarding the result of the analysis of their behaviour at the initial testing. [8] This paper is mainly focused on the following optimization methods: Differential Evolution and Downhill Simplex. The average success of these two methods has rapidly increased.

3.1. Differential Evolution and its proposed changes

The optimization method uses traditional evolutionary operators: crossover, mutation and selection. The method uses the principle of adaptive mutation parameter – the Ali and Törn adaptive rule. [22] A new individual is created by the mutation of the best individual with four randomly selected individuals (they are different from the best individual in the population) and the current selected element of the population – BEST method. [23] This principle keeps the diversity of search (diversification - exploration) at an early stage and increases the intensity of the search at a later stage (intensification - exploitation). The principle of differential evolution is based on the selection of a better individual from two individuals - parents and their offspring. The offspring is created by the crossing of parents and creates a new individual (created by the mutation of individuals) - BEST method. The better individual from these two compared individuals is subsequently inserted into the population. This population completely replaces the parent population.

To avoid premature convergence (duplication of a good individual to another population leads to faster finding of the solution but reduces the diversity and converges too early) we proposed the rule: If the difference between the objective function value of the current generated individual and the objective function of the best found individual is less than $1 \cdot 10^{-11}$ the gen of the individual to crossover with the parent is changed with 30 % probability. The gen is changed to a value which equals the current value of the coordinates + (+ with 50 % probability and – with 50 % probability) the value of the lowest step in the search space which the optimization method can perform * 10 * random number in the interval 0 (including) and less than 1 using uniform distribution.

3.2. Downhill Simplex and its proposed changes

This heuristic search method uses a set of n + 1 linearly independent candidate solutions (n denotes search space dimension) - Simplex. Ordinarily this method uses a set of n + 1 linearly independent candidate solutions (n denotes search space dimension) - Simplex. We deliberately break the rule of the linearly independent points. The optimization method could generate the same point in the search space. The success of finding the optimum was very small in this case. The method uses four basic phases – Reflection, Expansion, Contraction and Reduction. [17]

This optimization method uses the idea of the Nelder–Mead Downhill Simplex algorithm. [18] This method uses the rounding of coordinates of the point to the nearest feasible coordinates in the search space and this leads to deviation from the original direction in our case. We performed additional optimization experiments with smaller steps to test the success of finding the optimum in this case. The success was higher than the success with the largest steps. We proposed a slight modification of the reduction phase. If the same point (same coordinates of the point) is generated in the search space, each point coordinate will be changed with 50 % probability to a value which equals the current value of the coordinates + (+ with 50 % probability and – with 50 % probability) the value of the lowest step in the search space which optimization method can perform. After this modification, this method achieved success comparable to other favourite methods Evolution Strategy and Simulated Annealing in finding global optimum.

4. Evaluation

If we are interested in the behaviour of the implemented optimization method we have to consider that it is using the random generation of a new possible solution. Hence we have to replicate optimization experiments with the same settings of the optimization method parameters to reduce the risk of random behaviour.
Considering the number of simulation experiments we can divide the number of simulation experiments into three groups:

- Simulation experiment - simulation run of simulation model
- Optimization experiment - performed with concrete optimization method setting to find optimum of objective function)
- Series - replication of optimization experiments with concrete optimization method setting

Optimization experiments were repeated with the same settings of the optimization method parameters – series. The same conditions had to be satisfied for each optimization method, e.g. the same termination criteria, the same search space. If the optimization method has the same parameters as another optimization method, we set up both parameters with the same boundaries (same step, low and high boundaries)

Another problem is that the setting of the optimization method parameters deeply affect the optimization method behaviour. Hence we tested different settings of the optimization methods parameters – different series.

4.1. Optimization Methods Success

The first criterion is the value of not finding the known VTR (value to reach). This value is expressed by:

$$f_1 = \frac{s - n_{succ}}{s} \tag{7}$$

where *s* denotes the number of performed series, n_{succ} denotes the series where the VTR was found. Simulation runs of all possible settings of simulation model input parameters were performed. Average Method Success of Finding Optimum can be formulated as follows:

$$f_{avg} = \left(1 - \frac{\sum_{i=1}^{s} f_{1_i}}{s} \right) \cdot 100[\%] \tag{8}$$

We tested new modified and improved optimization methods on different simulation models – see Fig. 1. We can see that the success of finding the optimum of the performed series of Downhill Simplex and Differential Evolution has rapidly increased. Downhill Simplex method was also faster than the other tested optimization methods. We have to emphasize that the number of controls (simulation model input parameters) was not large. We performed more replications of optimization experiments with new methods than in the initial testing to improve the quality of optimization results. We performed tests for independence of the generated random numbers. We found that the random distribution was not uniform due to multiple initializing of the random-number generator (randomize function). Random Search optimization method was not so successful after this modification. Experimental results and experience show that Differential Evolution converges better than other stochastic algorithms. [18] We can agree with this statement after comparing the results of average success of finding the optimum shown in the following chart.

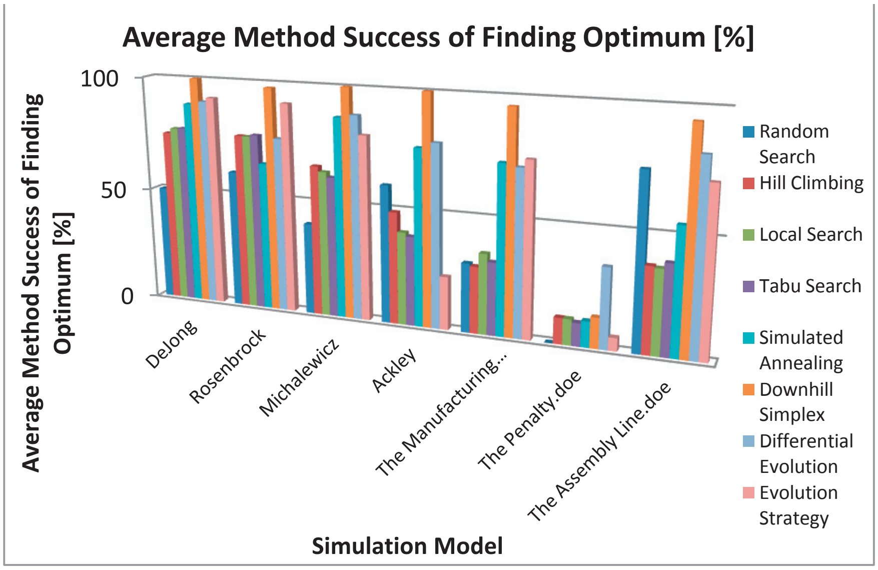

Fig. 1. Average Optimization Method Success – New Modified Optimization Methods.

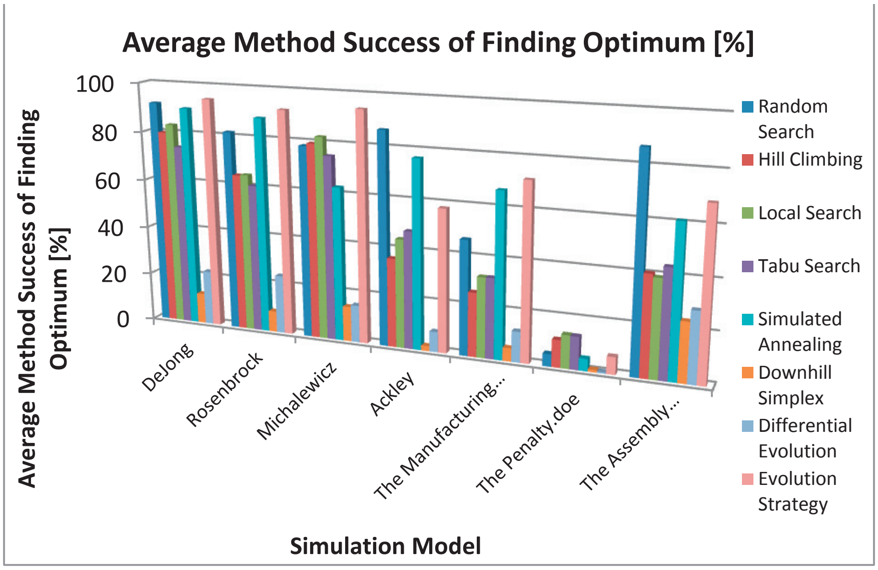

Fig. 2. Average Optimization Method Success – Old Optimization Methods.

4.2. Finding suitable settings of optimization method parameters for each simulation model

The first step of finding the suitable settings for each optimization method parameter was the evaluation of all series according to proposed evaluative functions. [8]

Furthermore, we selected only those series of the optimization where the value of the first criterion ("Number of Found Optima or Objective Function Values Close to Optimum") equals the best found value of this criterion from all tested series. We found that the lowest found value of the first criterion equals zero (this means at least one series had a 100% success for finding the global optimum from all the performed series of the optimization method). The criterion value is the number in the interval 0 ÷1 (including). This condition was selected because some of the optimization methods may not be able to find a global optimum with 100% success in the search space even in the best performed series. We subsequently counted the number of series with a 100% success rate (according to the first criterion) with a specific setting of optimization method parameter for each simulation model. This number was divided by the total number of series performed with the selected parameter value for a specific model, regardless of the success of a particular series. This calculated value expresses the percentage of series success. A series represents the specific parameter value of the optimization method.

We created a bar graph which expresses the success of a specific setting of the optimization method parameter. The height of the bar expresses the effect of the optimization parameter value on the optimization method success of finding the optimum. If these bars of the optimization method parameter are the same height we can say the optimization methods have little propensity to bad tuning of this parameter.

We found that some settings of optimization methods parameters have a little propensity to be a bad method for tuning parameters in the case of our tested simulation models, e.g. Evolution Strategy. We can see that the setting of reflection and contraction coefficient affects the method behaviour of finding global optimum more than the expansion and reduction coefficient for our simulation models – see Fig. 3, Fig. 4. We can see that some settings (series) of Reflection and Contraction Coefficient have a lower percentage of absolutely successful series with different simulation models (Rosenbrock; The Manufacturing System and Logistics model; The Assembly Line model).

a b

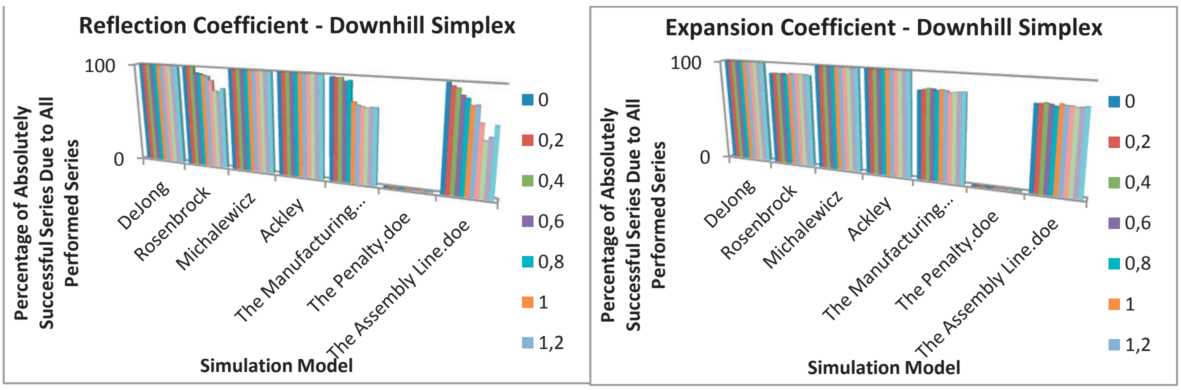

Fig. 3. (a) Reflection Coefficient – The Success of Downhill Simplex Series –Maximization of Values; (b) Expansion Coefficient – The Success of Downhill Simplex Series –Maximization of Values.

c d

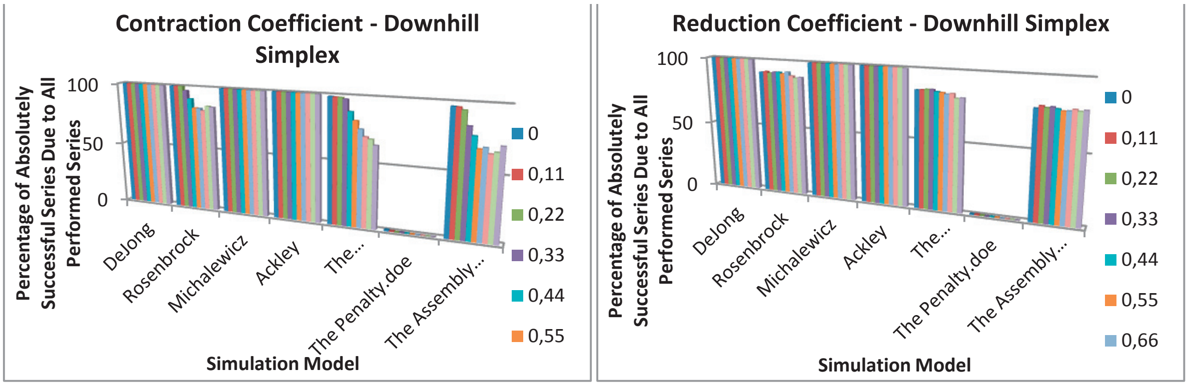

Fig. 4. (c) Contraction Coefficient – The Success of Downhill Simplex Series –Maximization of Values; (d) Reduction Coefficient – The Success of Downhill Simplex Series –Maximization of Values.

The following charts (see Fig. 5, Fig. 6) suggest that it is heavier to set the parameters of Differential Evolution. The success of these optimization methods depends on the objective function landscape.

a b

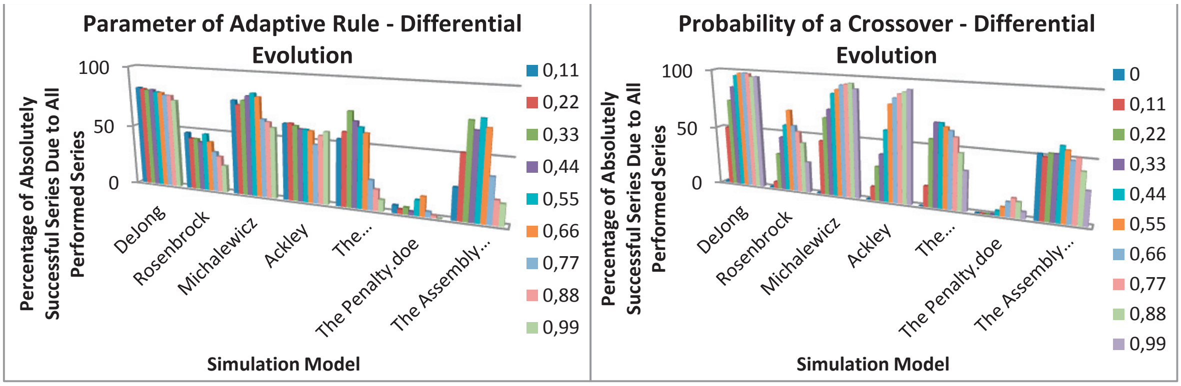

Fig. 5. (a) Parameter of Adaptive Rule – The Success of Differential Evolution Series –Maximization of Values; (b) Probability of a Crossover – The Success of Differential Evolution Series –Maximization of Values.

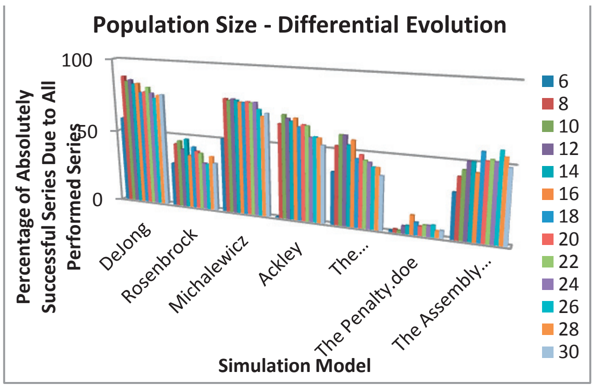

Fig. 6. Population Size – The Success of Differential Evolution Series –Maximization of Values.

Conclusion

The goal of the research is to compare selected optimization methods - Random Search, Stochastic Hill Climbing, Stochastic Tabu Search, Stochastic Local Search, Downhill Simplex, Simulated Annealing, Differential Evolution and Evolution Strategy. These optimization methods are used to search for the global optimum of the objective function. The success of heuristic optimization methods depends on the objective function landscape. Hence it is necessary to analyse the problem and try to imagine the objective function landscape (multimodal, roughly linear, continuous, etc.) Some optimization methods generate a whole population instead of one solution. This feature prevents premature convergence.

Evolution Strategy is a suitable optimization method for all the tested objective functions (little propensity to be a bad method for tuning parameters). The alternative to Evolution Strategy is Simulated Annealing. Simulated Annealing has the ability to escape from the local extreme thanks to the implemented approach of setting the temperature to the initial temperature (temperature expresses decreasing probability of acceptance of a worse solution). Optimization methods based on pseudo-gradient searching such as Stochastic Hill-Climbing, Stochastic Local Search and Stochastic Tabu Search achieve almost the same results for the simple objective function landscape due to their similar nature. We also tested the newly modified Differential Evolution method which avoids repressing the diversity of solutions (elitism). An advantage of this approach is the faster finding of a feasible solution but not the finding of the global optimum.

Very good results were achieved with the modified Downhill Simplex method. This optimization method works by calculating the points of the centroid (centre of gravity of the simplex). We used the rounding of coordinates of the vector to the nearest feasible coordinates in the search space. The proposed changes to Differential Evolution and Downhill Simplex in this paper have led to increasing the average success of finding the optimum in the search space by tens of percentage points.

Optimization methods achieved good success for finding the optimum due to the quite small dimension of the search space. We defined the quite small dimension of the search space regarding the simulation of all possible solutions to find the real optimum.

We would like to test the other listed optimization methods on discrete event simulation models with a higher number of controls (simulation model input parameters) and compare their efficiency.

Acknowledgements

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

References

[1] R. Schoech, S. Schmid, C. Hillbrand a R. Fleisch, Optimising plant layout decisions based on emulation models - technical framework and practical insights, International Journal of Simulation and Process Modelling, Vol. 8, No. 2-3, pp. 92-103, 2013.

[2] P. Horejsi, J. Polcar, Implementation of source engine for virtual tours in manufacturing factories, in: Advances in Visual Computing, Berlin, Springer Berlin Heidelberg, 2013, pp. 737-746.

[3] Arena, OptQuest for Arena, Rockwell Automation, 2014. Available: http://www.arenasimulation.com/Products_OptQuest.aspx. Accessed: 2014-01-03.

[4] WITNESS Optimizer, Lanner, Available: http://www.lanner.com/en/media/witness/optimiser.cfm.Accesed: 2014-05-03.

[5] Siemens, Plant Simulation, Siemens, 2014.Available: http://www.plm.automation.siemens.com/cz_cz/products/tecnomatix/plant_design/plant_simulation.shtml. Accessed: 2014-01-10.

[6] OR/MS, OR/MS TODAY, INFORMS - Institute for Operations Research and the Management Sciences, October 2013. Available: http://www.orms-today.org/surveys/Simulation/Simulation6.html. Accessed: 2014-01-03.

[7] P. Raska, Z. Ulrych, Testing optimization methods on discrete event simulation models and testing functions, Procedia Engineering, Vol. 69, pp. 768–777, 2014.

[8] P. Raska, Z. Ulrych, Simulation Optimizer and Optimization Methods Testing On Discrete Event Simulations Models and Testing Functions, in: Proceedings of the European Modeling and Simulation Symposium, ATHENS, GREECE, 2013.

[9] J. April, F. W. Glover, J. P. Kelly a M. Laguna, Simulation-based optimization: practical introduction to simulation optimization, in: WSC '03 Proceedings of the 35th conference on Winter simulation: driving innovation, 2003, pp. 71-78.

[10] C. A. Floudas a C. E. Gounaris, A review of recent advances in global optimization, Journal of Global Optimization, Vol. 45, No. 1, pp. 3-

38, 2009.

[11] L. Hong a B. L. Nelson, A brief introduction to optimization via simulation, in: 2009 Winter Simulation Conference, WSC 2009, Austin, 2009, pp. 75-85.

[12] F. Longo, On the short period production planning in industrial plants: a real case study, Int. J. Simulation and Process Modelling, Vol. 8, No. 1, pp. 17-28, 2013.

[13] D. Rani a M. M. Moreira, Simulation–Optimization Modeling: A Survey and Potential Application in Reservoir Systems Operation, Water Resources Management, Vol. 24, No. 6, pp. 1107-1138, 2010.

[14] C. Schaefer, Particle Algorithms for Optimization on Binary Spaces, ACM Transactions on Modeling and Computer Simulation, Vol. 23, No. 1, pp. 8:1-8:25, 2013.

[15] J. A. Nelder, R. Mead, A Simplex Method For Function Minimization, The Computer Journal, Vol. 1, No. 7, pp. 308-313, 1964.

[16] A. J. Monticelli, R. Romero, E. N. Asada, Fundamentals of Simulated Annealing, in: Modern Heuristic Optimization Techniques, IEE Press ed., M. E. El-Hawary, Ed., New Jersey, John Wiley & Sons, 2008, pp. 101-120.

[17] T. Weise, E-Book Global Optimization Algorithms - Theory and Application 2nd Edition, Available: http://www.it-weise.de/projects/book.pdf Accessed: 2011-02-02.

[18] J. Tvrdik, Evolutionary algorithms - Textbooks (in Czech language Evoluční algoritmy - učební texty), University of Ostrava, 2004. Available: http://prf.osu.cz/doktorske_studium/dokumenty/Evolutionary_Algorithms.pdf. Accessed: 2012-02-06.

[19] F. Koblasa, Evolution Algorithm for Job Shop Scheduling Problem Constrained by the Optimization Timespan, pp. 350-357, 2013.

[20] K. P. Wong, Z. Y. Dong, Differential Evolution, in: Modern Heuristic Optimization Techniques, M. E. El-Hawary, Editor, New Jersey, John Wiley & Sons, 2008, pp. 171–186.

[21] V. Miranda, Fundamentals of Evolution Strategies and Evolutionary Programming, in: Modern heuristic optimization techniques, M. E. El-Hawary, Editor, New Jersey, John Wiley & Sons, 2008, pp. 43–60.

[22] M. M. Ali, A. Törn, Population set-based global optimization algorithms: some modifications and numerical studies, Computers & Operations Research, Vol.31, No. 10, pp. 1703–1725, September 2004.

[23] R. Storn a K. Price, A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, in Journal of Global Optimization, 1997.