Self-Organizing Migrating Algorithm Applied to Discrete Event Simulation Optimization

Pavel Raska
Zdenek Ulrych
Department of Industrial Engineering – Faculty of Mechanical Engineering
University of West Bohemia
Univerzitni 22, Pilsen 306 14, Czech Republic
praska@kpv.zcu.cz
ulrychz@kpv.zcu.cz

Abstract

The paper deals with testing and evaluation of a modified Self-Organizing Migrating Algorithm (SOMA) applied to different discrete event simulation models. These models are focused on real problems in industrial companies. The SOMA heuristic optimization method is derived from the Differential Evolution method, which is effective for different dimensional search spaces of the simulation models. We specify the ranges of each algorithm parameter and test all the possible combinations of settings of the algorithm within the specified ranges. We repeat the simulation optimization experiments to reduce the random behaviour of the algorithm. We propose a methodology using different evaluation criteria to analyse the SOMA behaviour of finding the optimum of an objective function specified for each discrete event simulation model.

Keywords
SOMA, Self-Organizing Migrating Algorithm, Discrete Event Simulation Models, Simulation Optimization and Evaluation.

1. Introduction

Most managers know that an industrial company is a complex and tightly connected system affected by many external and internal factors. Managers have to make many decisions in a good way and without bad consequences. A possible answer to quality management is the use of discrete event simulations to test different scenarios without the need for physical implementation in a company. We can use a digital replica of physical assets, processes, people, etc. – digital twins (Krajcovic et al. 2018). The other advantage of using simulation experiments is preventing problems, e.g. manufacturing optimization – identification of the bottlenecks, low utilization of resources etc.; logistics optimization – idle time, shortening shipping time; warehousing – low amount or large amount of supplies, etc. (Buckova et al. 2018), which may occur, and which we cannot imagine due to the high complexity of the simulated system. Discrete event simulation in connection with optimization – simulation optimization - can be more effective for management because it can avoid bad human decision-making and show the efficiency of different possible scenarios (suitable solutions to the modelled problem) that can be implemented in a real system. A suitable solution represents the feasible settings of the simulated system under specified constraints. The search space is usually boundary-constrained, and is defined as follows:

$$\tilde{X} = \prod_{j=1}^{n} [a_j, b_j], a_j \leq b_j \tag{1}$$

Where $\tilde{X}$ denotes the search space; n denotes the dimension of the search space (number of the input parameters of the discrete event simulation model); a_j denotes the lower bound of the interval of j-th decision variable (input parameter of the discrete event simulation model); b_j denotes the upper bound of the interval of j-th decision variable.

Simulation optimization is a technique for finding the global optimum of the optimized objective function respecting the specified constraints. The equation reflects a process of finding a possible solution using experimentation with a discrete event simulation model reflecting the modelled problems.(Weise 2009),(Zelinka 2016)

$$\breve{\mathbf{X}} = \operatorname{argmin}_{\mathbf{X}\in\widetilde{X}} F(\mathbf{X}) = \{\breve{\mathbf{X}} \in \widetilde{X}: F(\breve{\mathbf{X}}) \leq F(\mathbf{X}) \forall \mathbf{X} \in \widetilde{X}\} \tag{2}$$

Where $\breve{\mathbf{X}}$ denotes the global minimum of the objective function (function maximization can be converted to function minimization); $F(\mathbf{X})$ denotes the objective function value of the candidate solution – the range includes real numbers, i.e. $F(\mathbf{X}) \subseteq \mathbb{R}$; $\widetilde{X}$ denotes the search space.
We use the simulation optimizer with different (heuristic, metaheuristic, etc.) optimization methods that vary the input parameters of the discrete event simulation model to find the optimal feasible solution of the modelled problem. The simulation optimizer provides a suitable setting or a list of suitable settings for the decision variables of a simulation model. The problem is that we cannot evaluate each possible solution – candidate solution - in most cases because of the huge number of possible solutions in the search space (NP hard problem). Another problem is that we cannot determine if the optimization method has found the global or local optimum of the objective function according to the termination criteria of the simulation experiments. We propose a methodology using different evaluation criteria to analyse the SOMA behaviour of finding the optimum of an objective function specified for each discrete event simulation model.

2. Discrete Event Simulation Models

We tested the behaviour of selected optimization methods on different discrete event simulation models reflecting real problems in industrial companies. We tested all the solution candidates of the search space of the discrete event simulation model to find the global optimum of the model's search space. This optimum represents the best specific settings of the simulation model input parameters – decision variables. The quality of this solution is represented by the global extreme of the model's objective function. Let's briefly describe these models.

2.1 The Production and Control Stations Model

All the discrete event simulation models were built in Tecnomatix Plant simulation software. The Production and Control Stations Model is focused on a production workshop – see Figure 1.

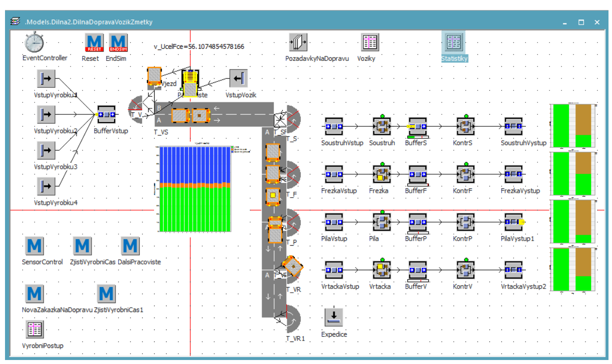

Figure 1. Production and Control Stations Model

The simulation model simulates six different workplaces. Four types of products are processed in the workshop. The time of the arrival of each product at the workshop is specified. The product passes through the workplaces respecting the technological processes. A forklift truck transports the products between workplaces.
The main goal is to determine the number of machines and controllers at individual workplaces according to the number of forklift trucks, machines and controller utilization (maximizing production processes). (Raska and Ulrych 2018)
The objective function:

$$F(\mathbf{X}) = \frac{\text{NumberOfProcessedProducts}}{10} + \sum_{WN=1}^{4} (\text{MachineUtilization} + \text{ControllerUtilization}) \tag{3}$$

2.2 The Transport Model

The simulation model describes the transport from the warehouse to the production lines by tractors. This model illustrates a situation where supply requirements are gradually collected. A tractor with trailers conveys the parts to the production lines from the warehouse at regular intervals. The transports are performed according to the requirements of the production lines. Each tractor has defined places to serve and this list of places does not change during the simulation. All supply requirements arise stochastically. The aim of the simulation study is to find the correct sequence of served places at the production lines. The transport time is shorter than the processing time of the supplied parts at the production/assembly line. The places for unloading the parts are indexed and the tractor has to pass through these places in ascending order – see Figure 2.

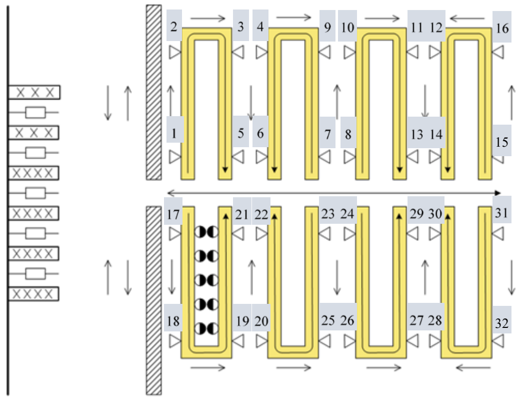

Figure 2. Diagram of the Transportation Path

The simulation model has eight production lines. Each production line has four places for unloading. Four tractors convey the parts from the warehouse. Each tractor pulls a maximum of four trailers. A trailer can convey a maximum of six different boxes with components. Two hundred different parts are conveyed from the warehouse to the production lines. Each tractor can supply from zero to ten places for unloading. Each tractor exits the warehouse every fifteen minutes.

The objective function reflects the average utilization of tractors with trailers conveying the small and large parts, and the finished product. The objective function also reflects the overall average utilization of the production lines. The average utilization of production lines in the objective function is superior to the average utilization of all types of tractors using the coefficients. The objective function is minimized.

Definitions of the objective functions:

$$F_1(\mathbf{X}) = \sum_{i=1}^{4} DT_i \tag{4}$$

$$F_2(\mathbf{X}) = \frac{\sum_{i=1}^{32} TO_i}{100} \tag{5}$$

$$F_3(\mathbf{X}) = \left(\sum_{i=1}^{200} NT_i\right) * 10000 \tag{6}$$

$$F(\mathbf{X}) = F_1(\mathbf{X}) + F_2(\mathbf{X}) + F_3(\mathbf{X}) \tag{7}$$

Where DT_i denotes the distance travelled of the i-th tractor [m]; TO_i denotes the time over the maximum delivery time of the part at the i-th place for unloading at the production line [sec]; NT_i denotes the number of transport requests to the i-th part to a production line where the part does not have the assigned tractor for the transport [sec]; $F(\mathbf{X})$ denotes the objective function. (Raska and Ulrych 2018)

2.3 AGV Transport Model

This practical discrete event simulation model deals with supplying the production lines using automated guided vehicles (AGVs). Large parts are supplied by the trailers and it is not possible to load a big number of these parts to satisfy the needs of the production line for a longer time. Hence more trailers must be used for the transport at once. It is also not possible to control the supply in a way that if the supply falls below a certain level there would be generated a requirement for transport from the warehouse (except for a limited number of some parts). This is caused by the transport time which is longer than the time of consumption of the parts transported to the production line.

The whole system of supplying the production lines is based on a simple principle: the tractor with the trailers continually transports the parts and after the unloading of transported parts goes to the warehouse or to pre-production for new parts and then transports them immediately to the production lines. Limited capacity of the buffer (parts storage) on the production line is a regulation in this case. Each tractor has a defined path using different loading and unloading stations which must be passed. The various types of parts are loaded and unloaded at different stations in the company. The parts can be loaded on the trailer at the loading stations in the warehouse or at the various production departments in the company. Each production line has several unloading stations for various parts. A schematic layout of the loading and unloading stations is shown in Figure 3. The decision variables are the number of each AGV type. The following illustrates a situation where a collapse of the whole transport system occurred (because of the blockage of one AGV by another AGV) – see Figure 3.

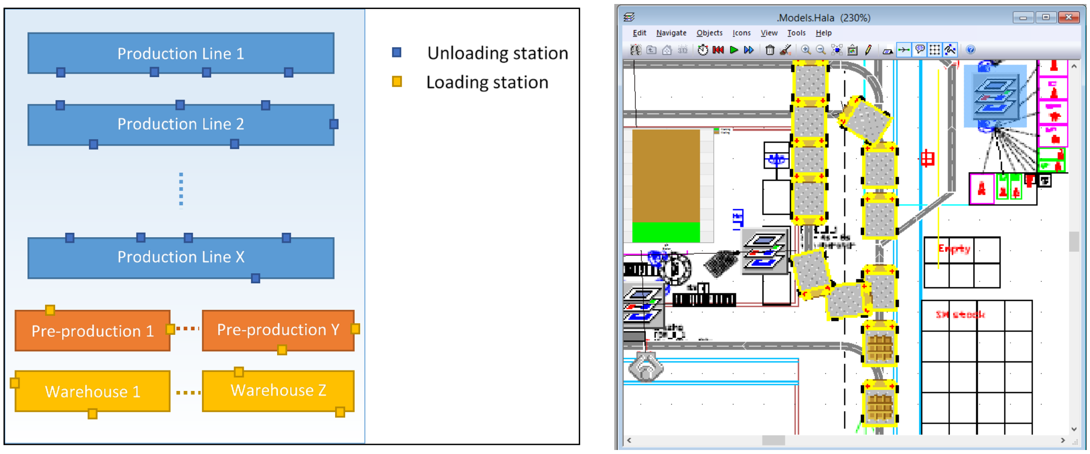

Figure 3. Simple Layout of Loading/Unloading Stations for AGV and Sample of AGV Collapse in the Simulation Model

The objective function reflects the average use of AGV (tractor with trailers). The objective function also reflects the overall average utilization of the production lines. The average use of the production lines is superior to the average use of the trains using the coefficients in the objective function. The objective function was maximized. (Pavel Raska and Ulrych 2014)
The objective functions definitions:

$$F_1(\mathbf{X}) = \sum_{i=1}^{n} VL_i \tag{8}$$

Where VL_i denotes the utilization of the *i*-th production line; n denotes the number of production lines.

$$F_2(\mathbf{X}) = \frac{\sum_{i=1}^{m}(10 - VTP_i)}{1000} \tag{9}$$

Where VTP_i denotes the number of AGV of the same type; m denotes the number of AGV of different types.

$$F(\mathbf{X}) = F_1(\mathbf{X}) + F_2(\mathbf{X}) \tag{10}$$

Where $F(\mathbf{X})$ denotes the resulting objective function.

3. Self-Organizing Migrating Algorithm - SOMA

SOMA is based on the self-organizing behaviour of groups of individuals in a ‘social environment’. It can also be classified as an evolutionary algorithm, despite the fact that no new generations of individuals are created during the search. Only the positions of the individuals in the search space are changed during a generation, called a ‘migration loop’. Individuals are generated at random according to what is called the ‘specimen of the individual’ principle. The specimen is in a vector, which comprises an exact definition of all these parameters that together lead to the creation of such individuals, including the appropriate constraints of the given parameters. SOMA is not based on the

philosophy of evolution (two parents create one new individual – the offspring), but on the behaviour of a social group of individuals.(Zelinka 2016)
The SOMA optimization method is derived from Differential Evolution. There are different modifications of the Differential Evolution e. g. (Elsayed et al. 2013), (Li et al. 2015).
The original source code in different programming languages can be downloaded at (Zelinka 2005).

3.1 The SOMA parameters

The *Mass* parameter denotes how far the currently selected individual stops from the leader individual (if the $Mass = 1$ then the currently selected individual stops at the position of the leader, if the $Mass = 2$ then the currently selected individual stops behind the position of the leader, which equals the distance of the initial position of the currently selected individual and the position of the leader). If the $Mass < 1$ then the currently selected individual stops in front of the leader which leads to degradation of the migration process (the algorithm finds only local extremes). Hence it is recommended to use $Mass > 1$. It is also recommended to use the following lower and upper boundary of the parameter $Mass \in [1.1,3]$.
The *Step* parameter denotes the resolution of mapping the path of the currently selected individual. It is possible to use a larger value for this parameter to accelerate the searching of the algorithm if the objective function is unimodal (convex function, few local extremes, etc.). If the objective function landscape is not known it is recommended to use a low value for this parameter. The search space will be scanned in more detail and this increases the probability of finding the global extreme. It is also important to set the *Step* parameter in a way that the distance of the currently selected individual and the leader is not an integer multiple of this parameter (the diversity of the population is reduced because each individual could be pulled to the leader and the process of searching for the optimum could stop at a local extreme). Hence it is recommended to use $Step = 0.11$ instead of $Step = 0.1$. The setting of e.g. $Step = 0.11$ also rapidly increases the effectiveness of SOMA Strategy All To All.
The *PRT* parameter denotes the perturbation. The Perturbation vector contains the information whether the movement of the currently selected individual toward the leader should be performed. It is one of the most important parameters of this optimization method and it is very sensitive. It is recommended to use $PRT = 0.1$. If the value of this parameter increases, then the convergence of the SOMA algorithm to local extremes also rapidly increases. It is possible to set this parameter to $PRT \in [0.7,1.0]$ if many individuals are generated and if in the dimension of the search space the objective function is low. If $PRT = 1$ then the stochastic part of the behaviour of SOMA is cancelled and the algorithm behaves according to deterministic rules (local optimization of the multimodal objective function).
The *NP* parameter denotes how many individuals are generated in a population. If this parameter is set to $NP = 2$ the SOMA algorithm behaves like a traditional deterministic method.
Generally if *n* (where *n* denotes the dimension of the search space) is a higher number, then this parameter can be set to $NP = [0.2,0.5] \times n$. If the objective function landscape is simple we can use a lower number of generated individuals. If the objective function is complicated we can set this parameter $NP = n$. It is recommended to use $NP \geq 10$.
This parameter is equivalent to the 'Generation' parameter used in other evolutionary algorithms. This parameter denotes the number of population regenerations. (Zelinka 2016), (Zelinka et al. 2013), (Volna 2012)

3.2 Strategies

There are several variants of the basic SOMA optimization method – strategies. The principle is to distinguish the cooperation of the individuals and the migration of the population in the search space. The strategies are: AllToOne – all the individuals in the population migrate to the leader, except the leader. The leader remains at its position during a migration loop - strategy index: 0; AllToAll – in this strategy, there is no leader. All individuals move towards the other individuals. The individual comes back to the best found solution after finishing the *NP*-1 individual migrations. This strategy is more time consuming, but the probability of finding the global extreme is higher, because the individuals can explore a bigger area of the search space - strategy index: 1; AllToAllAdaptive – this strategy is similar to AllToAll strategy. The difference between the AlltoAll strategy is that individuals do not begin a new migration form the same old position, but from the last best position found during the last migration to the previous individual - strategy index: 2; AllToRand – this is a strategy where all individuals move towards a randomly selected individual during the migration loop, no matter what cost value this individual has. It is up to the user to decide how many randomly selected individuals there should be - strategy index: 3. (Zelinka 2016)

4. Simulation experiments

A simulation run performed on a discrete event simulation model can be time consuming, depending on the difficulty of the discrete event simulation model. We have to repeat the simulation optimization experiments to reduce the random behaviour of the optimization method. We can divide the number of the simulation experiments as follows: Simulation experiment – simulation run of the simulation model; Optimization experiment – performed with a specific optimization method setting to find the optimum of the objective function; Series – replication of optimization experiments with a specific optimization method setting. This replication ensures the reduction of the influence of random implemented in the optimization algorithm. We tested different 2,304 settings of the SOMA method. We defined a step and lower and upper boundaries for these parameters – see Table 1. We designed a simulation optimizer based on Client - Server architecture to reduce the time of the optimization experiment. This architecture allows simulation optimization experiments to be performed on many remote simulation optimizers - servers. These servers use the local database of the simulation experiments loaded to computer memory, or an external database of simulation experiments which is stored on the Client computer.

Table 1. Settings of SOMA Parameters

Parameter	Step	Lower Bound	Upper Bound
Mass	0.5	1.1	2.6
Step	0.4	0.11	1.31
PRT	0.1	0.1	0.6
NP	1×n	1×n	6×n
Migrations	10	10	10
Strategy index (strategy type)	1	0	3

4.1 Database of Simulation Experiments

We tested all the possible solution candidates inside the search space of the discrete event simulation model on remote optimizers in the initial stage. After mapping, we created databases containing solution candidates - settings of the simulation model input parameters - and the objective function values for each discrete event simulation model. The simulation optimizer downloads this database of solution candidates to local computer memory. Each solution candidate - the possible settings of the simulation model input parameters – and its objective function values, is encoded into one number to accelerate the searching of the database. The simulation optimizer does not have to perform the simulation run in simulation software, but it only searches for the solution candidate in the internal memory of all the solution candidates. If the local server or the client external database does not contain the solution candidate, the simulation optimizer performs the simulation experiment with the settings of the simulation model input parameters – the candidate solution - and calculates its objective function value. Then it saves the solution candidate and its objective function value in its own local server and in the external database of the client. This user option increases the speed of the simulation optimization experiment.

4.2 Termination Criteria

The same termination criteria were satisfied for each optimization method. We specified the first termination criterion – Value to Reach - because we mapped all the solution candidates in the search space and we know the best solution of the modelled problem. This best solution candidate represents the global optimum of the objective function. If the optimization method finds a solution candidate whose objective function value is within the defined tolerated deviation ($\varepsilon = 0.001$ in our case) from the objective function value of the global optimum, the optimization experiment is stopped.

The second termination criterion is the maximum number of simulation runs that the simulation optimizer can perform in the optimization experiment for each model. We performed many optimization experiments in the initial stage of testing and we confirmed that the settings of the optimization method could significantly affect the performance of the optimization method. Hence we tested many different settings of the optimization methods to reduce the number of bad settings of the optimization methods parameters and to reduce the random nature of the selected optimization methods (each of the selected methods uses random distribution).

We calculated this maximum number using information entropy - Shannon Entropy. The number of all possible solutions in the search space is reduced using information entropy. (Borda 2011)

The reduction coefficient:

$$\delta = \max\{0{,}1 - \beta \cdot log\tilde{X}\}, \delta \in [0{,}1] \tag{11}$$

Where $\tilde{X}$ denotes the size of the search space – the number of all possible solutions in the search space; β denotes the coefficient of search space reduction.

The maximum number of simulation runs that it is possible to perform in each optimization experiment – the second termination criterion:

$$\tilde{X}_H = \lfloor 2^{\delta \cdot log_2 \tilde{X}} \rfloor \tag{12}$$

The following table shows the specifications of the tested discrete event simulation models (Table 2). This table also contains the maximum number of optimization experiments performed using information entropy – reducing the search space.

Table 2. Specifications of the Discrete Event Simulation Models

Discrete Event Simulation Model	$F(\check{\mathbf{X}})$	$F(\hat{\mathbf{X}})$	n	$\tilde{X}$	$\tilde{X}_H$
The Transport Model	44,502.25	4.26E+07	8	252,000,000	74,512
The Production and Control Stations Model	2.149858	57.29105	9	259,200,000	74,847
AGV Transport Model	0.1368983	9,1	15	14,515,200	39,558

The curve in the chart (Figure 4) shows the dependence of the second termination criterion on the number of the possible solutions in the search space of the discrete event simulation model. The second termination criterion is not much reduced if the number of the possible solutions in the search space is small. We set the coefficient $\beta = 0.05$ according to our initial optimization experiments.

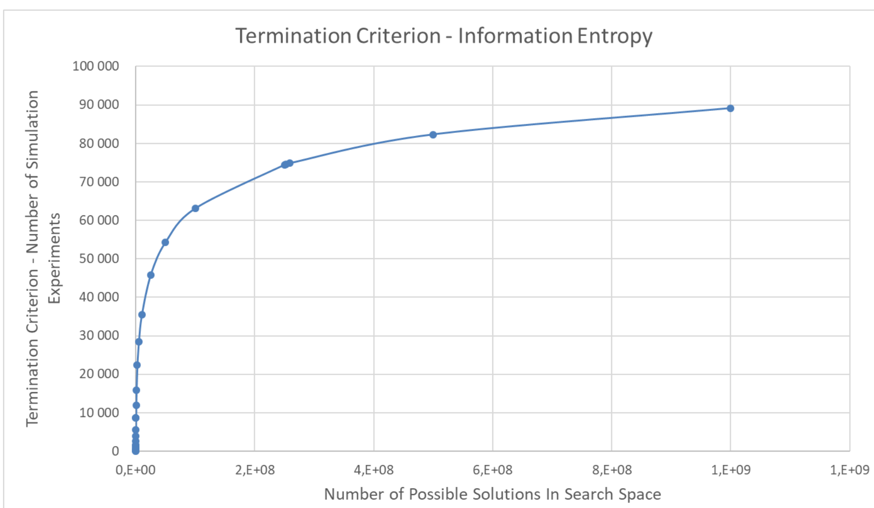

Figure 4. Termination Criterion - Information Entropy

5. The Frequency of Using the Database Records

We have developed a methodology that evaluates each optimization experiment to evaluate the quality of the behaviour of the optimization method. This methodology uses different evaluation criteria (e.g. finding the optimum, the rate of convergence to the optimum, the range of the purpose function values in the simulation experiments, etc.) for each setting of the tested optimization method. A more detailed description of the proposed methodology can be found in paper (Raska and Ulrych 2018).

We mapped all feasible solutions and their objective function values from all discrete event simulation models, so we found the lower and upper bounds of the objective function of each simulation model.

$$F(\mathbf{X}) \in [F(\check{\boldsymbol{X}}), F(\hat{\boldsymbol{X}})], F(\check{\boldsymbol{X}}) < F(\hat{\boldsymbol{X}}), F(\mathbf{X}) \in \mathbb{R}, \boldsymbol{X} \in \tilde{X} \tag{13}$$

Where $F(\mathbf{X})$ denotes the objective function value of feasible solution $\mathbf{X}$; $F(\check{\boldsymbol{X}})$ denotes the objective function value of the found global minimum in the Search Space $\tilde{X}$; $F(\hat{\boldsymbol{X}})$ denotes the objective function value of the found global maximum in the Search Space; $\mathbb{R}$ denotes the real numbers. We divided this whole interval of objective function values into 100 smaller parts - intervals ($n = 100$) with the same step:

$$\varepsilon = \frac{|F(\hat{\boldsymbol{X}}) - F(\check{\boldsymbol{X}})|}{n} \tag{14}$$

Where n denotes the number of smaller intervals of the objective function values of feasible solutions with the defined size of the interval - ε.
We counted the relative frequencies of the feasible solutions objective function values between the specified range (we calculated how often the objective function values occur within different ranges of objective function values):

$$FR_j, j \in [1, n] = \begin{cases} FR_j + 1 \text{ if } \left(F(\boldsymbol{X}_i) \geq \varepsilon * (j-1) + F(\check{\boldsymbol{X}})\right) \wedge \left(F(\boldsymbol{X}_i) < F(\check{\boldsymbol{X}}) + \varepsilon * (j-1) + \varepsilon\right), i \in [1, NCS] \\ FR_j + 0 \quad \text{else} \end{cases} \tag{15}$$

Where FR_j denotes the frequency of the feasible solutions objective function values; j denotes the index of the small interval of the objective function value, $F(\boldsymbol{X}_i)$ denotes the objective function value of the i-th feasible solution belonging to interval of the objective function; NCS denotes the number of feasible solutions in the interval of the objective function value; ε denotes the size of the range of the smaller interval.

$$PRF_j = \left(\frac{FR_j}{NO}\right) * 100[\%], j \in [1, n] \tag{16}$$

Where PRF_j denotes the percentage relative frequency of the j-th interval; NO denotes the number of occurrences of the objective function value belonging to the whole interval.
The following figures show the percentage of the calculated relative frequencies of the mapped objective function values of the discrete event simulation model considering the intervals of the objective function values – see Figure 5, Figure 6, Figure 7. These data series reflect the quality of the feasible solutions of the modelled problem.

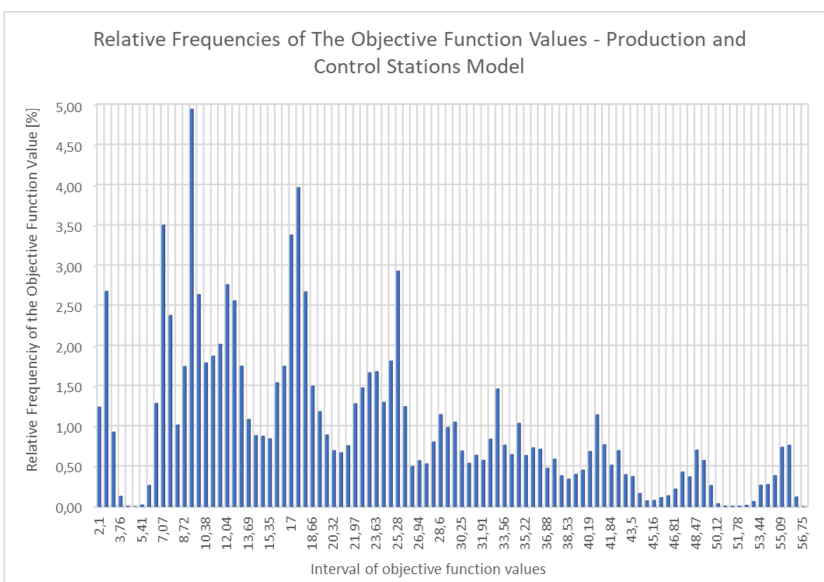

Figure 5. Relative Frequencies of The Objective Function Values - Production and Control Stations Model – Objective Function Maximization

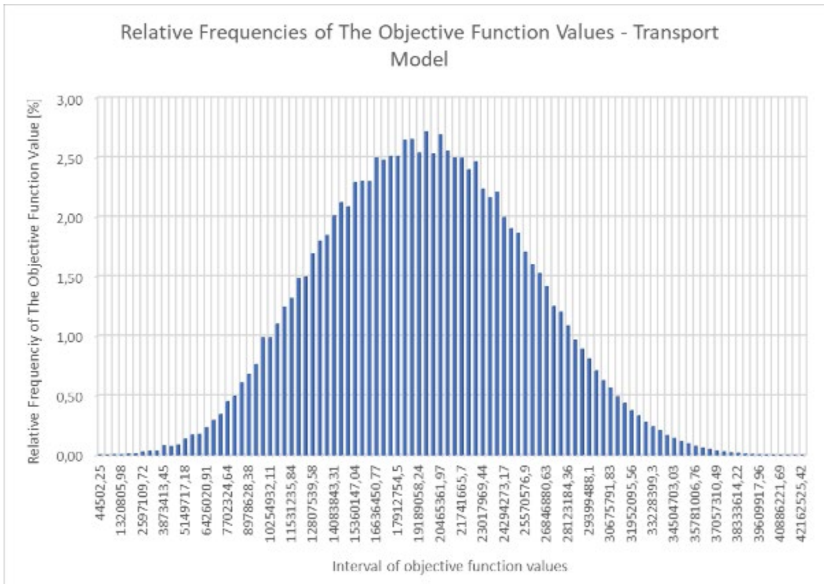

Figure 6. Relative Frequencies of The Objective Function Values - Transport Model – Objective Function Minimization

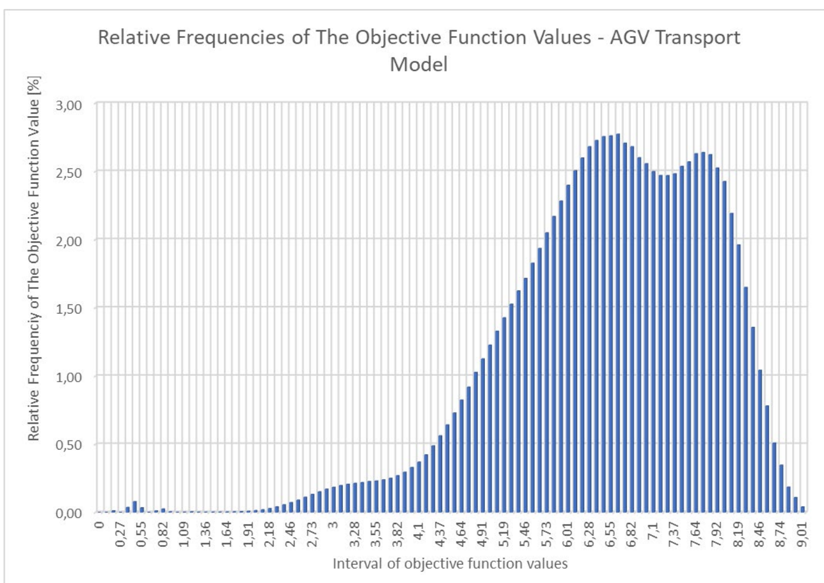

Figure 7. Relative Frequencies of The Objective Function Values – AGV Transport Model – Objective Function Maximization

We calculated the quality of each series - tested the settings of the optimization method - which comprises all the proposed evaluation criteria. We calculated the quartile characteristics – the minimum, the first quartile, the median, the third quartile and the maximum of the whole range of values representing the quality – the weighted sum for all the proposed criteria - a more detailed description of the evaluation criteria can be found in the paper (Raska and Ulrych 2018). We tested these series to obtain and compare their quality due to their relative frequency of using the database records. Other charts show the percentage of relative frequencies of the objective function values found by the optimization method – see Figure 8, Figure 9, Figure 10.

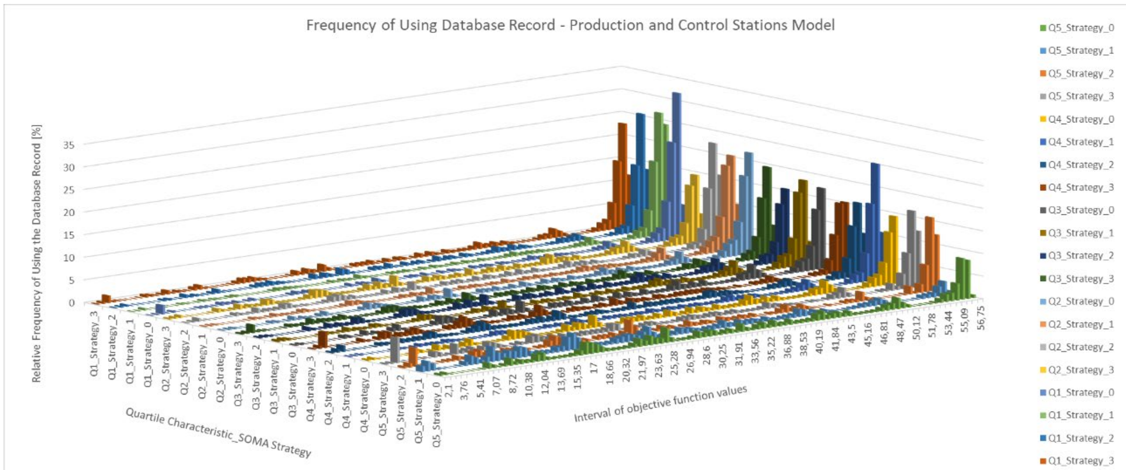

Figure 8. Frequency of Using Database Record - Production and Control Stations Model – Objective Function Maximization

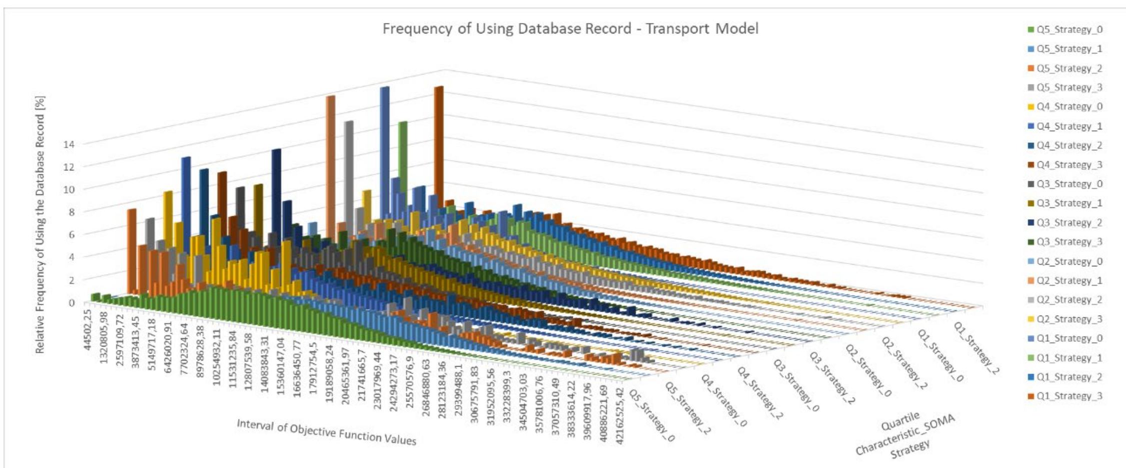

Figure 9. Frequency of Using Database Record - Transport Model – Objective Function Minimization

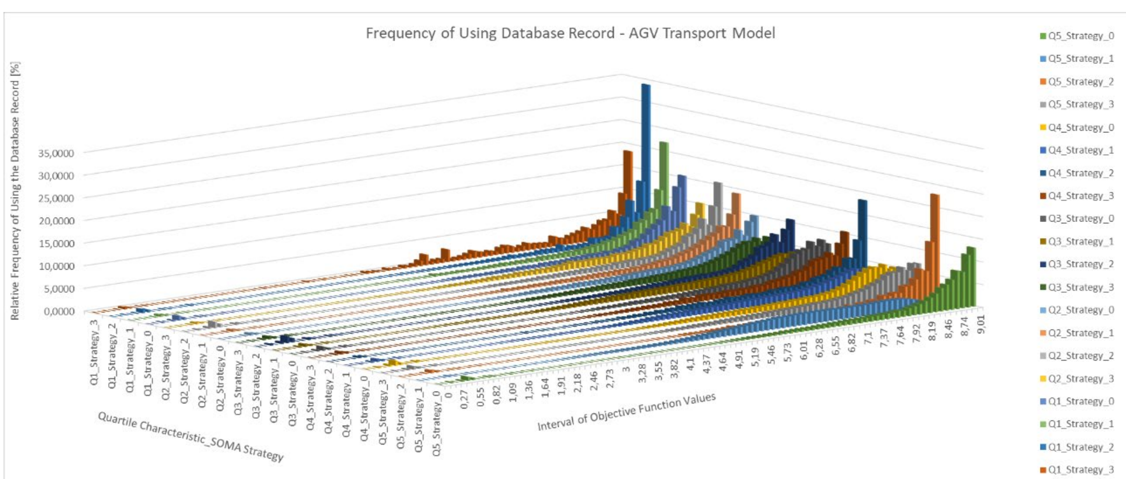

Figure 10. Frequency of Using Database Record - AGV Transport Model – Objective Function Maximization

If we compare the relative frequencies of the mapped objective function values of the discrete event simulation model and the relative frequencies of the objective function values found by the optimization method, the strategies strongly tend to converge to the global optimum. If we select less suitable optimization method settings the relative proportion starts to increase in other areas of the interval – see Figure 8, Figure 9, Figure 10. Appropriate setting of the optimization method leads to a lower number of optimization experiments to find the global optimum (a steeper convergence to the global optimum and the lower relative frequency in the area of global extremes of the objective function values).

A small relative frequency around the optimum is due to the selected type of termination criteria. A possible criterion of termination is a Value to Reach - if the candidate solution is found (its objective function value is equal to the objective function value or the objective function value is within the defined tolerated deviation from the objective function value of the global optimum), the simulation optimization experiment is stopped. Each series – repetition of optimization experiments with concrete settings of the optimization method - was evaluated according to different evaluation criteria. One of the evaluation criteria is Optimization Method Success - finding the global optimum (the candidate solution whose objective function value is within a defined tolerance of the global optimum objective function). Figure 11 shows the comparison of the success of each SOMA strategy from the best to the worst settings of the optimizing method for one selected model (with the highest differences of the objective function values).

The AllToOne strategy (index 0) is prone to wrong optimization method parameters settings in our case of discrete event simulation models. This strategy is followed by the SOMA strategy AllToRand (index 3), the AllToAllAdaptive strategy (index 2) and the worst strategy AllToAll (index 1) to set the wrong optimization method parameters.

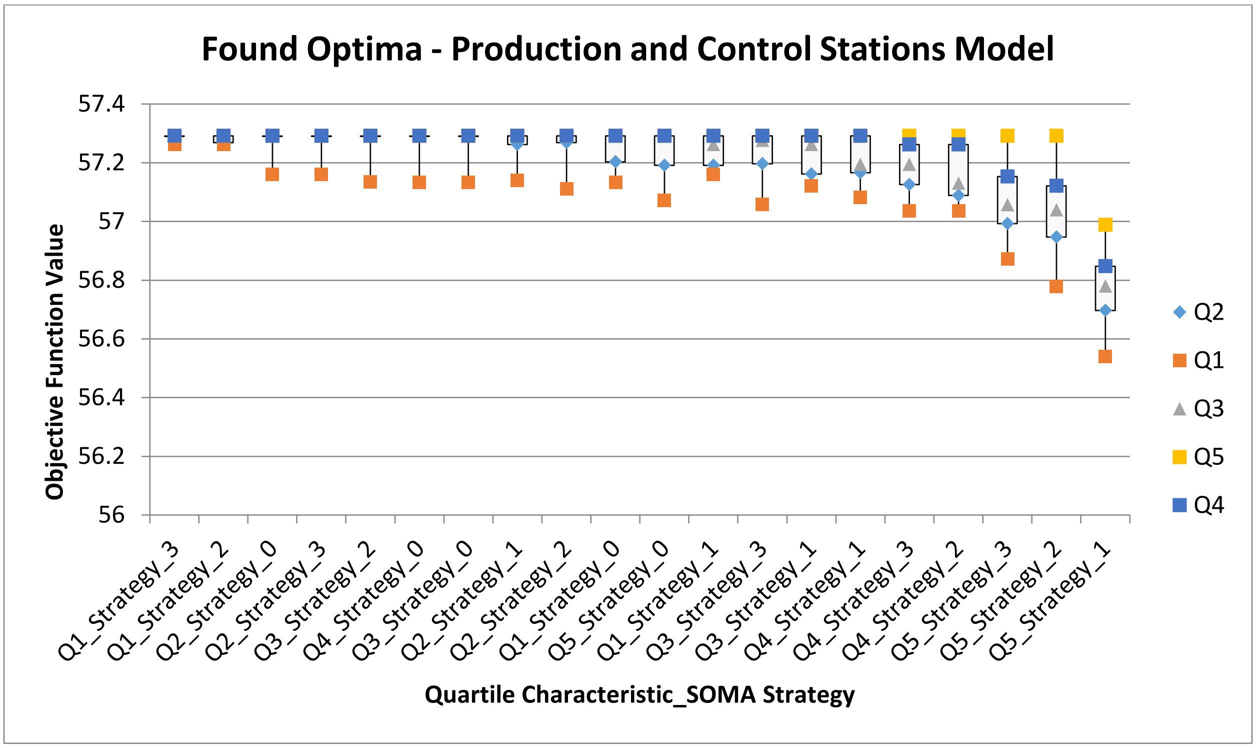

Figure 11. Found Optima - Production and Control Stations Model – Objective Function Maximization

6. Conclusion

The testing of the SOMA optimization method confirmed that it is a quite successful and generic optimization method according to different objective function landscapes of our tested discrete event simulation models. This method found the global optimum or an acceptable solution even in the case of the worst settings of this optimization method. This method is derived from Differential Evolution but some SOMA strategies provide worse solutions than Differential Evolution. We tested all the SOMA strategies under the same conditions of the simulation experiments – the same termination criteria, number of repetitions in the series, and the same setting of the basic parameters of this optimization method. The AllToAll strategy of SOMA is not useful when the second termination criterion is set hard – the user sets a small number of simulation experiments to perform. We tested AllToAll strategy with a higher number

of simulation experiments in the series. This SOMA strategy was better if we specified a softer second termination criterion – SOMA can perform more simulation experiments.
If we compare the SOMA strategies, the AllToOne strategy is prone to wrong optimization method parameters settings and is able to find the global optimum of the objective function, or the local optimum near to the global optimum of the objective function in the case of our tested discrete event simulation models. If we sort the quality of SOMA strategies from best to worst according to the versatility of using the strategy when the user can perform a limited number of experiments, the order is: AllToOne, AllToRand, AllToAllAdaptive and the worst strategy is AllToAll.

Acknowledgements
This paper was created with the subsidy of the project SGS-2018-031 "Optimizing sustainable production system parameters" carried out with the support of the Internal Grant Agency of the University of West Bohemia.

References
Borda, M., *Fundamentals in Information Theory and Coding*, Springer Berlin Heidelberg, Berlin, 2011.
Buckova, M., Krajcovic, M., Plinta, D., Use of dynamic simulation in warehouse designing, *2nd International Conference on Intelligent Systems in Production Engineering and Maintenance, ISPEM 2018*, Wroclaw; Poland; September 17 – 18,2018.
Elsayed, S., Sarker, R., Essam, D., An Improved Self-Adaptive Differential Evolution Algorithm for Optimization Problems, *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 89-99, 2013.
Krajcovic, M., Grznar, P., Fusko, M., Skokan, R., Intelligent logistics for intelligent production systems, *Communications - Scientific Letters of the University of Zilina*, vol. 20, no. 4, pp. 16-23, 2018.
Li, Y., Zhan, Z., Gong, Y., Chen, W., Zhang, J., Li, Y., Differential Evolution with an Evolution Path: A DEEP Evolutionary Algorithm, *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1798-1810, 2015.
Raska, P., Ulrych, Z.,Testing Different Evolution Strategy Selection Strategies, *MM Science Journal*, vol. 2018, no. 1, pp. 2290-2299, 2018.
Raska, P., Ulrych, Z., Hierarchical Approach to Developing a Logistic Discrete Event Simulation Model Using Automated Guided Vehicles, *26th European Modeling and Simulation Symposium, EMSS 2014*, Bordeaux, France, September 10 - 12, 2014.
Volna, E., Evolutionary Algorithms and Neural Networks,. http://www1.osu.cz/~volna/Evolucni_algoritmy_a_neuronove_site.pdf, 2012.
Weise, T., Global Optimization Algorithms–Theory and Application, http:// http://www.it-weise.de/projects/bookNew.pdf, December 7, 2011.
Zelinka, I., *SOMA—Self-Organizing Migrating Algorithm,* Studies in Computational Intelligence, Springer Verlag, 2016.
Zelinka, I., Snasel, V., Abraham, A., Handbook of Optimization: From Classical to Modern Approach, *Intelligent Systems Reference Library,* vol. 2013, no. 38, 2013.
Zelinka, I., SOMA - Source Codes, http://www.ft.utb.cz/people/zelinka/soma/, 2005.

Biography / Biographies
Pavel Raska is a Doctor at the Department of Industrial Engineering and Management at the University of West Bohemia in Pilsen (Czech Republic). He holds M.Sc., and Ph.D. in Mechanical Engineering at the same university. His research interests are oriented towards discrete event simulation, simulation optimization, modelling and simulation tools (ARENA, Plant Simulation) and working on practical simulation projects for companies.

Zdenek Ulrych is Associate Professor at the Department of Industrial Engineering and Management at the University of West Bohemia in Plzen and he is also a researcher in the Regional Technological Institute at the University of West Bohemia in Plzen (Czech Republic). He holds M.Sc., Ph.D. and doc. in Mechanical Engineering at the same university. His research interests are oriented towards discrete event simulation, optimization in simulation, modelling and simulation tools (ARENA, Plant Simulation), design and development of software and working on practical simulation projects for companies.