DOI: https://doi.org/10.46354/i3m.2019.emss.047

EVALUATION OF A SELF-ORGANIZING MIGRATING ALGORITHM APPLIED TO DISCRETE EVENT SIMULATION OPTIMIZATION

Pavel Raska[(a)], Zdenek Ulrych[(b)]

[(a)] Department of Industrial Engineering - Faculty of Mechanical Engineering, University of West Bohemia,
Univerzitni 22, 306 14 Pilsen
[(b)] Department of Industrial Engineering - Faculty of Mechanical Engineering, University of West Bohemia,
Univerzitni 22, 306 14 Pilsen

[(a)]praska@kpv.zcu.cz, [(b)] ulrychz@kpv.zcu.cz,

ABSTRACT

The paper deals with testing and evaluation of a modified Self-Organizing Migrating Algorithm (SOMA) applied to a discrete event simulation model reflecting the supply of production lines using automated guided vehicles. The SOMA heuristic optimization method is derived from the Differential Evolution method. We test all the SOMA strategies under the same conditions of the simulation experiments – the same termination criteria, number of repetitions in the optimization experiments, and the same setting of the basic parameters of the SOMA. We propose a methodology using different evaluation criteria to analyse the different SOMA strategies 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 Model, AGV, Evaluation

1. INTRODUCTION

Many of today's industrial companies are large complex systems affected by a host of internal and external factors. An important thing is effective management of company resources and processes. We can use a digital replica of a physical system – a digital twin. This simulation model can answer the question "What happens if …". The simulation of different scenarios can avoid bad human decision-making and also prevent system errors in the company before they occur e.g. timely identification of bottlenecks, mapping and increasing the utilization of company resources, effective scheduling and resource allocation, bin packing problems (Koblasa, Vavrousek, and Manlig 2017), transport distances (Bučková, Krajčovič, and Edl 2017), etc. Another problem is we cannot often calculate and evaluate each possible scenario because of the large number of possible solutions in the search space – NP hard problem. The search space is usually boundary-constrained and it represents the domain of the input parameters of the discrete event simulation model. The search space is defined as follows:

$$\tilde{X} = \prod_{j=1}^{n} \tilde{X}_j = \prod_{j=1}^{n} [a_j, b_j], a_j \leq b_j, a_j, b_j \in \mathbb{R} \qquad (1)$$

Where symbols denote: $\tilde{X}$ denotes the search space; j denotes the j-th decision variable of the simulation model; n denotes the dimension of the search space; a_j denotes the lower bound of the interval of j-th decision variable; b_j denotes the upper bound of the interval of j-th decision variable.

The goal of using the simulation optimization is to find the best quality solution (represented by the objective function value) according to the other possible solutions.

We proposed the simulation optimizer with different (heuristic, metaheuristic, etc.) optimization methods that vary the discrete event simulation model input parameters to find the optimal solution of the modelled problem. The possible solution containing the settings of each simulation model input parameter is defined as follows:

$$\mathbf{X}[j] = x_j \forall j : j = \{1, 2, \dots, n\} \qquad (2)$$

where $\mathbf{X}[j]$ denotes a possible solution – the vector of the values for each decision variable; x_j denotes the value of the j-th decision variable; j denotes the index of the decision variable.

If we minimize the objective function values, the global optimum is defined as follows:

$$\check{\mathbf{X}} = \mathrm{argmin}_{\mathbf{X} \in \tilde{X}} F(\mathbf{X}) = \{\check{\mathbf{X}} \in \tilde{X} : F(\check{\mathbf{X}}) \leq F(\mathbf{X}) \forall \mathbf{X} \in \tilde{X}\} \ (3)$$

where $\check{\mathbf{X}}$ denotes the global minimum of the objective function; $F(\mathbf{X})$ denotes the objective function value of the candidate solution – the range includes real numbers, i.e. $F(\mathbf{X}) \subseteq \mathbb{R}$; $\tilde{X}$ denotes the search space.

The function maximization can be converted to function minimization by multiplying the objective function value by -1.

The simulation optimizer provides a candidate solution (best found feasible solution of the modelled problem) or a list of candidate solutions. The problem is that we cannot confirm that the candidate solution provided by the optimization method is the real optimum of the objective function because we cannot evaluate all possible solutions in the search space - NP hard problem. Another problem is that the optimization method is prone to bad settings of its parameters. Hence, we proposed a methodology using different evaluation criteria to analyse the SOMA behaviour of finding the optimum of an objective function.

2. 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, even though 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, Sarker, and Essam 2013; Li et al. 2015)

The original source code in different programming languages of the tested SOMA can be downloaded at (Zelinka 2005).

2.1. The SOMA parameters

The *Mass* parameter denotes how far the currently selected individual stops from the leader individual (if $Mass = 1$ then the currently selected individual stops at the position of the leader, if $Mass = 2$ 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 $Mass < 1$ 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. (Volna 2012; Zelinka 2016; Zelinka et al. 2013;

2.2. Strategy of Individual Movements

There are several variants of the basic SOMA optimization method – strategies of individual movements. 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)

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: a tractor with trailers continually transports the parts and after the unloading of the transported parts it goes to the warehouse or to pre-production for new parts and then transports them immediately to the production lines – see Figure 1.

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 2. 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 2.

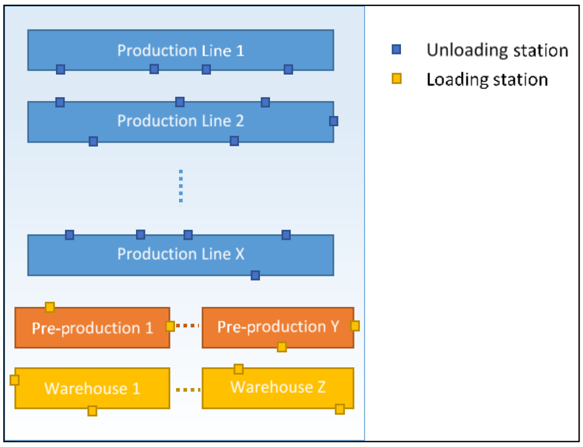

Figure 1: Simple Layout of Loading/Unloading Stations for AGV

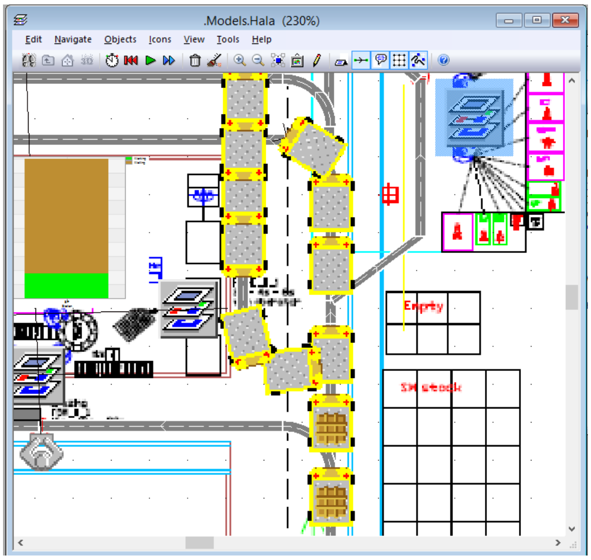

Figure 2: Sample of AGV Collapse in the Simulation Model

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 average use of the trains using the coefficients in the objective function. The objective function is maximized. (Raska and Ulrych 2014)

The objective functions definitions:

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

where U_i denotes the utilization of the i-th production line; n denotes the number of production lines.

$$F_2(\mathbf{X}) = \frac{\sum_{i=1}^{m}(10 - \ N_i(\mathbf{X}))}{1000} \tag{5}$$

where N_i denotes the number of AGV of the same type; m denotes the number of AGV of different types.

$$F(\mathbf{X}) = F_1(\mathbf{X}) + F_2(\mathbf{X}) \tag{6}$$

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

4. OPTIMIZATION EXPERIMENTS

We have to repeat the simulation optimization experiments. This replication ensures the reduction of the influence of random implemented in the optimization algorithm. We 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

We tested 2,304 different settings of the SOMA optimization method. Table 1 shows the defined step and lower and upper boundaries for the SOMA parameters.

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

We designed our own simulation optimizer based on Client - Server architecture. This architecture allows us to reduce the time of the optimization experiment because the series - replication of optimization experiments with a specific optimization method setting - can be performed on many remote simulation optimizers - servers.

We tested all the items - possible solutions - of the discrete event simulation model search space. We selected only feasible solutions. These feasible solutions represent the feasible settings of the simulation model input parameters under specified constraints. We created a database containing these solutions and their objective function values. Each feasible solution and its objective function value is encoded into one number to accelerate the searching of the database. A remote simulation optimizer on the servers downloads this database from the client to local computer memory before performing the simulation runs. The simulation optimizer does not have to perform the simulation run in simulation software, but it only searches for the possible solution in the internal memory. If the local memory on the server does not contain the possible solution, the optimizer asks for this solution from the client's external database. All the servers are connected to this remote database and if the local memory of the simulation optimizer does not contain this new generated possible solution by the optimizer, the simulation optimizer performs the simulation experiment with the settings of the simulation model input parameters and calculates its objective function value. Then it saves the possible solution 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.

4.1. Termination Criteria

The same termination criteria were satisfied for each tested series. We specified the first termination criterion – Value to Reach - because we mapped all the solution candidates in the search space. The found value of Value to Reach is the global optimum/maximum of the objective function (the objective function maximization):

$$\widehat{\boldsymbol{X}} = \text{argmax}_{\mathbf{X} \in \tilde{X}} F(\mathbf{X}) = \{\widehat{\boldsymbol{X}} \in \tilde{X} : F(\widehat{\boldsymbol{X}}) \geq F(\mathbf{X}) \forall \mathbf{X} \in \tilde{X}\} \tag{7}$$

where $\widehat{\boldsymbol{X}}$ denotes the global maximum of the objective function; $F(\widehat{\boldsymbol{X}})$ denotes the objective function value of the global maximum; $F(\mathbf{X})$ denotes the objective function value of the possible solution – the range includes real numbers, i.e. $F(\mathbf{X}) \subseteq \mathbb{R}$; $\tilde{X}$ denotes the search space.

Because we know the global optimum of the objective function we could specify the tolerated deviation ($\varepsilon = 0.001$ in our case) from the objective function value of the global optimum. The optimization methods also stop searching for the global optimum if they find a possible solution whose objective function value is within the defined tolerance.

$$|F(\mathbf{X}) - F(\widehat{\boldsymbol{X}})| \leq \varepsilon \tag{8}$$

where $F(\mathbf{X})$ denotes the objective function of the found possible solution in the optimization experiment with concrete settings of the optimization method parameters; $F(\widehat{\boldsymbol{X}})$ denotes the objective function value of the global maximum.

The second termination criterion is the maximum number of simulation runs that the simulation optimizer can perform in the optimization. We performed many optimization experiments in the initial stage of testing.

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. Another reason for repeating the optimization experiments is reduction of the random nature of the optimization method (SOMA 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:

$$\delta = \max\{0{,}1 - \beta \cdot log\tilde{X}\}, \delta \in [0{,}1] \tag{9}$$

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.

$\tilde{X}_H$ denotes the maximum number of simulation runs that the optimization method can perform in each optimization experiment. It is defined as follows:

$$\tilde{X}_H = \lfloor 2^{\delta \cdot log_2 \tilde{X}} \rfloor \tag{10}$$

The following table shows the specifications of the tested discrete event simulation model – the global minimum, the global maximum, the dimensions of the search space, the number of possible solutions in the search space, the maximum number of simulation runs that the simulation optimizer can perform in the optimization using information entropy. We set the coefficient $\beta = 0.05$ according to our initial optimization experiments – see Table 2.

Table 2: Specifications of the Discrete Event Simulation Model

Discrete Event Simulation Model	$F(\breve{\mathbf{X}})$	$F(\hat{\mathbf{X}})$	n	$\tilde{X}$	$\tilde{X}_H$
AGV Transport Model	0.1368983	9,1	15	14,515,200	39,558

5. EVALUATION CRITERIA

Many research papers use the average or the standard deviation to evaluate the performance of the optimization methods. This evaluation criterion is sufficient for commonly used testing functions, e.g. De Jong's, Rosenbrock's, Ackley's function, etc. (Pohlheim 2006)
We proposed different evaluation criteria which express the success or the failure of the optimization method in different ways. Each criterion value is between [0, 1]:

$$f_i \in [0{,}1] \forall i : i = \{1{,}2, \ldots, 5\} \tag{11}$$

If the failure is 100[%], the criterion equals 1, therefore we try to minimize all specified evaluation criteria. The second and the third criteria are calculated from the Box plot characteristics - the smallest observation – sample minimum Q_1, lower quartile Q_2, median Q_3, upper quartile Q_4, and largest observation - sample maximum Q_5. These characteristics are calculated for each performed series – the setting of the optimization method parameters.

We calculated the quality of each series 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 other proposed criteria. The following boxplot charts show the main criteria calculated for each series sorted according to the calculated weighted sum of all the proposed criteria:

$$f = f_1 \cdot w_{f_1} + f_2 \cdot w_{f_2} + f_3 \cdot w_{f_3} + f_4 \cdot w_{f_4} + f_5 \cdot w_{f_5} \tag{12}$$

where f denotes the weighted sum of all criteria; f_1 denotes the value of the first criterion – The Optimization Method Success; w_{f_1} denotes the weight of the first criterion; f_2 denotes the value of the second criterion - The Difference Between Optimum and Local Extreme; w_{f_2} denotes the weight of the second criterion; f_3 denotes the value of the third criterion - The Distances of Quartiles; w_{f_3} denotes the weight of the third criterion; f_4 denotes the value of the third criterion - The Number of Simulation Experiments Until the Optimum Was Found; w_{f_4} denotes the weight of the fourth criterion; f_5 denotes the value of the fifth criterion - Convergence to the Optimum (objective function values of the possible solutions generated in series by the optimization algorithm).
Table 3 shows the value specified for each weight. The sum of the weights equals one.

Table 3: Specifications of The Weights for Each Criterion

Weight	**Value**
w_{f_1}	0.35
w_{f_2}	0.25
w_{f_3}	0.2
w_{f_4}	0.15
w_{f_5}	0.05

5.1. Optimization Method Success

If we know the global minimum and the global maximum in the search space we can calculate the range of the objective function values:

$$F_{\tilde{X}} = \left| F(\breve{\mathbf{X}}) - F(\hat{\mathbf{X}}) \right| \tag{13}$$

where $F(\breve{\mathbf{X}})$ denotes the objective function value of the global minimum in the search space; $F(\hat{\mathbf{X}})$ denotes the objective function value of the global maximum in the search space.
The problem is to define the global minimum or maximum where we cannot test all the possible solutions in the search space – NP-hard problem. This is a common situation in industrial simulation optimization. We can only calculate the difference between the objective function value of the found best solution candidates in the search space in all series (we cannot confirm that the

found best solution candidate is the global optimum) - and the objective function value of the worst found possible solution of the search space (the maximum if the objective function is minimized):

$$\Delta F_{\tilde{X}} = F(\mathbf{X}^*) - F(X_{Worst}) \tag{14}$$

where $F(\mathbf{X}^*)$ denotes the objective function value of the found best candidate solution of the search space in all series; $F(X_{Worst})$ denotes the objective function value of the worst found possible solution (element) of the search space.

The value of the first criterion represents the failure of finding the best solution candidates in the search space in one series performed by the optimization method – value minimization. This criterion is expressed by Pseudo Pascal code – see Figure 3:

```
begin
  $n_{Succ} \longleftarrow 0$;
  (*browse all the items of the list*)
  for $i \longleftarrow 0$ to Length($X^*$) $- 1$ do
      (*optimum or acceptable candidate solution was found*)
      if $|F(X^*[i]) - F(\mathbf{X}^*)| \leq \varepsilon$ then
          $n_{Succ} \longleftarrow n_{Succ} + 1$;
      (*standardization - % share of unsuccessful series*)
      result $\longleftarrow \frac{\text{Length}(X^*) - n_{Succ}}{\text{Length}(X^*)}$;
end;
```

Figure 3: Pseudo Pascal Algorithm of the First Criterion Calculated for One Series – Finding the Global Optimum or Suboptimum

where X^* denotes the list of found best feasible solutions - candidate solutions – for all the optimization experiments performed in one series; Length(X^*) denotes the length of the list of the candidate solutions - number of found global/local optima with the concrete settings of the optimization method parameters; $\mathbf{X}^*$ denotes the found best candidate solution (global maximum if we mapped all the possible solutions in the search space) in all series; ε denotes the tolerated deviation from the value of the objective function value of the found best candidate solution in the search space in all series; $F(\,)$ denotes the objective function value; n_{Succ} denotes the counter of successful finding of the found best candidate solutions in the search space in all series.

The average method success of finding the best solution candidates in the search space from all the series (if we know the optimum) can be formulated as follows:

$$f_{1AVG} = \left(1 - \frac{\sum_{i=1}^{s} f_{1_i}}{s}\right) \cdot 100[\%] \tag{15}$$

where f_{1AVG} denotes the average method success of finding the optimum of the objective function (criterion maximization), i denotes the index of one series; f_{1_i} denotes the standardised scalar value of the i-th series (the optimization method failure – minimization of the criterion), s denotes the number of performed series.

We calculated the average success of all the SOMA strategies of finding the optimum of the objective function – see Figure 4.

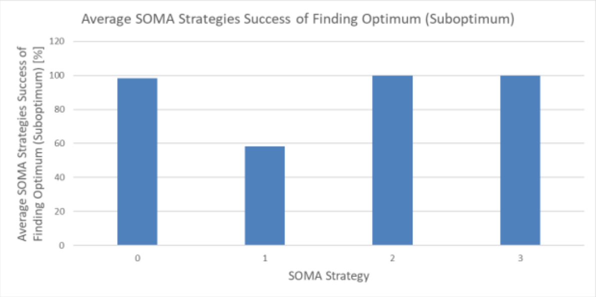

Figure 4: Average SOMA Strategies Success of Finding Optimum (Suboptimum)

We calculated the quality of each series which comprises all the proposed evaluation criteria using the weighted sum as the main criterion. We sorted these series using their values of the weighted sum. We calculated the minimum, the first quartile, the median, the third quartile and the maximum of the whole interval of the weighted sums.

We selected the series that represent the calculated boxplot characteristics. The next box plot chart shows the success of each SOMA strategy from the worst to the best series (concrete settings of the SOMA optimization method parameters) – see Figure 5.

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

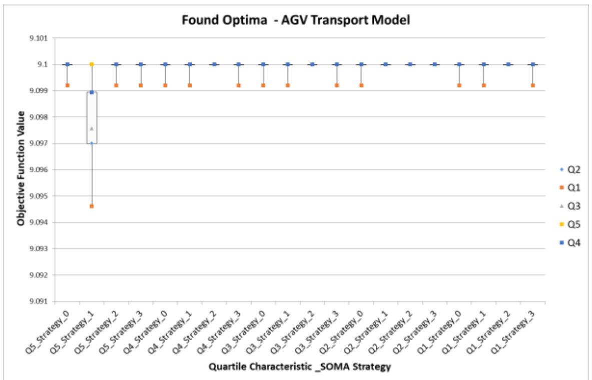

Figure 5: Found Optima for Different Strategies of Individual Movements - Objective Function Maximization

5.2. The Difference between Optimum and Local Extreme

The second criterion is useful when there is no series which contains any optimum or a best solution whose objective function value is within the tolerance of the optimum objective function value. The first criterion equals zero in this case. The output of the function can take value $f_2 \in [0,1]$. This function evaluates the difference between the objective function value of the found best solution in the series and the optimum of the objective function value. The task is to minimize this evaluation function. The list of found optima considering objective function value is sorted in ascending order. This sorting can be performed using the comparator function, which compares the quality (objective function values) of two possible solutions. If we maximize the objective function:

$$\mathrm{CF}_{F(\mathbf{X})}(\mathbf{X}_1,\mathbf{X}_2) = \begin{cases} -1 & \text{if } F(\mathbf{X}_1) > F(\mathbf{X}_2) \\ 1 & \text{if } F(\mathbf{X}_1) < F(\mathbf{X}_2) \\ 0 & \text{else} \end{cases} \tag{16}$$

where $\mathrm{CF}_{F(\mathbf{X})}$ denotes the comparator function comparing the objective function values of two possible solutions; $F(\mathbf{X}_1)$ denotes the objective function value of first possible solution.

After that the value of the second criterion is calculated using the formula:

$$f_2 = \left(\frac{F(\mathbf{X}^*) - F(X_{Best})}{F_{\tilde{X}}}\right) \tag{17}$$

where $F(\mathbf{X}^*)$ denotes the objective function value of the optimum of the search space in all series; $F(X_{Best})$ denotes the objective function value of the best solution candidate found in a concrete series; $F_{\tilde{X}}$ denotes the difference between the objective function value of the global minimum and the global maximum in the search space.

5.3. The Distances of Quartiles

The third criterion expresses the distance between the quartiles of a concrete series. The output of the function can take value $f_3 \in [0,1]$. If the first criterion equals zero then the third criterion equals zero – an absolutely successful series. The task is to minimize this evaluation function. Weights are used for evaluation purposes. These weights penalize the solutions placed in quartiles. The values of the weights are defined based on the results of the simulation experiments. The user can define the weight value. The sum of weights equals one. The third criterion when the objective function is minimized can be formulated as follows:

$$f_3 = \frac{f_{3w1} + f_{3w2} + f_{3w2} + f_{3w4} + f_{3w5}}{F_{\tilde{X}}} \tag{18}$$

$$f_{3w5} = |Q_1 - F(\mathbf{X}^*)| \tag{19}$$

$$f_{3w4} = w_{4_{f_3}}|Q_1 - Q_2| \tag{20}$$

$$f_{3w3} = w_{3_{f_3}}|Q_2 - Q_3| \tag{21}$$

$$f_{3w2} = w_{2_{f_3}}|Q_3 - Q_4| \tag{22}$$

$$f_{3w1} = w_{1_{f_3}}|Q_4 - Q_5| \tag{23}$$

Where $F(\mathbf{X}^*)$ denotes the objective function value of the global optimum of the search space; $w_{4_{f_3}}$ denotes the weight (penalty) of objective function values between sample minimum Q_1 and lower quartile Q_2; $w_{3_{f_3}}$ denotes the weight of objective function values between lower quartile Q_2 and median Q_3; $w_{2_{f_3}}$ denotes the weight of objective function values between median Q_3 and upper quartile Q_4; $w_{1_{f_3}}$ denotes the weight of objective function values between upper quartile Q_4 and largest observation - sample maximum Q_5.

If the objective function is maximized the parts of the third criterion are calculated as follows:

$$f_{3w5} = |F(\mathbf{X}^*) - Q_5| \tag{24}$$

$$f_{3w4} = w_{4\mathfrak{f}_3}|Q_4 - Q_5| \tag{25}$$

$$f_{3w3} = w_{3\mathfrak{f}_3}|Q_3 - Q_4| \tag{26}$$

$$f_{3w2} = w_{2\mathfrak{f}_3}|Q_2 - Q_3| \tag{27}$$

$$f_{3w1} = w_{1\mathfrak{f}_3}|Q_1 - Q_2| \tag{28}$$

Table 4 shows the value specified for each weight for the third, fourth and the fifth criterion The sum of the weights equals one.

Table 4: Specifications of Weights for The Third, Fourth and The Fifth Criterion

Weight	Value
$w_{1\mathfrak{f}_i}$	0.05
$w_{2\mathfrak{f}_i}$	0.1
$w_{3\mathfrak{f}_i}$	0.25
$w_{4\mathfrak{f}_i}$	0.6

5.4. The Number of Simulation Experiments Until the Optimum Was Found

The fourth criterion evaluates the speed of finding the optimum – the number of performed simulation experiments until the optimum/best solution was found in each series. The output of the function can take the value $f_4 \in [0,1]$. The task is to minimize this evaluation function. The fourth criterion when the objective function is minimized or minimized can be formulated as follows:

$$f_4 = \frac{f_{4w1} + f_{4w2} + f_{4w2} + f_{4w4} + f_{4w5}}{m_{\tilde{X}}} \tag{29}$$

$$f_{4w5} = |Q_1 - 1| \tag{30}$$

$$f_{4w4} = w_{4_{f_4}}|Q_1 - Q_2| \tag{31}$$

$$f_{4w3} = w_{3_{f_4}}|Q_2 - Q_3| \tag{32}$$

$$f_{4w2} = w_{2_{f_4}}|Q_3 - Q_4| \tag{33}$$

$$f_{4w1} = w_{1_{f_4}}|Q_4 - Q_5| \quad (34)$$

The next box plot shows the number of simulation experiments until the optimum (suboptimum) was found. The task is to minimize the chart values – see Figure 6. High chart values of the strategy AllToAll (index 1) show that this strategy has a problem to find the optimum of the objective function. The leader of the fastest finding of the optimum is AllToOne strategy (index 0).

The AllToRand (index 3) is a little bit faster than the AllToAllAdaptive strategy (index 2).

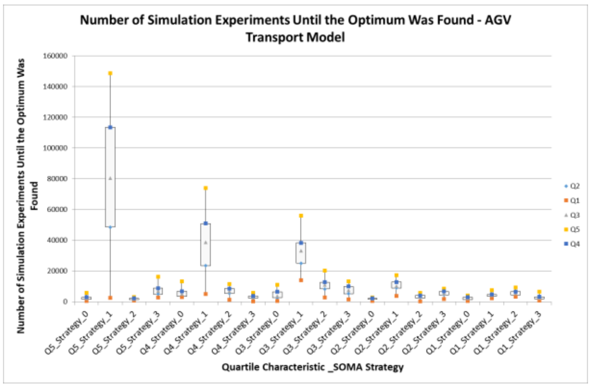

Figure 6: Number of Simulation Experiments Until the Optimum Was Found – Value Minimization

The AllToAll strategy (index 1) is not useful when the second termination allows a small number of simulation experiments to be performed. We tested this strategy with a higher number of simulation experiments, which SOMA can perform in the series. This strategy was better than other strategies of SOMA.

5.5. Convergence to the Optimum

The fifth criterion evaluates the convergence to the optimum. We store the objective function values of all feasible solutions generated by the optimization method in the optimization experiments in the series. The output of the function can take the value $f_5 \in [0,1]$. The task is to minimize this evaluation function. The fifth criterion when the objective function is minimized can be formulated as follows:

$$f_5 = \frac{f_{5w1} + f_{5w2} + f_{5w2} + f_{5w4} + f_{5w5}}{F_{\tilde{X}}} \quad (35)$$

$$f_{5w5} = |Q_1 - F(\mathbf{X}^*)| \quad (36)$$
$$f_{5w4} = w_{4\mathrm{f}_5}|Q_1 - Q_2| \quad (37)$$
$$f_{5w3} = w_{3\mathrm{f}_5}|Q_2 - Q_3| \quad (38)$$
$$f_{5w2} = w_{2\mathrm{f}_5}|Q_3 - Q_4| \quad (39)$$
$$f_{5w1} = w_{1\mathrm{f}_5}|Q_4 - Q_5| \quad (40)$$

If the objective function is maximized, the parts of the fifth criterion are calculated as follows:

$$f_{5w5} = |F(\mathbf{X}^*) - Q_5| \quad (41)$$
$$f_{5w4} = w_{4\mathrm{f}_5}|Q_4 - Q_5| \quad (42)$$
$$f_{5w3} = w_{3\mathrm{f}_5}|Q_3 - Q_4| \quad (43)$$
$$f_{5w2} = w_{2\mathrm{f}_5}|Q_2 - Q_3| \quad (44)$$
$$f_{5w1} = w_{1\mathrm{f}_5}|Q_1 - Q_2| \quad (45)$$

The next box plot shows the boxplot characteristics of the intervals of the objective function values of the generated feasible solution by the SOMA strategies. The task is to maximize the chart values – see Figure 7.

The leaders of the process of generating quality solutions are the AllToAllAdaptive strategy (index 2) and the AllToOne strategy (index 0) in our model. The AllToRand strategy (index 3) is worse than the previous strategies. The low chart values of the strategy AllToAll (index 1) show that this strategy has a problem to generate quality individuals in a small number of simulation experiments. This strategy supports the exploration and it needs to perform more simulation experiments to find the global optimum.

In the context of optimization, exploration means finding new points in areas of the search space which have not been investigated before. 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 novel 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.(Weise 2009)

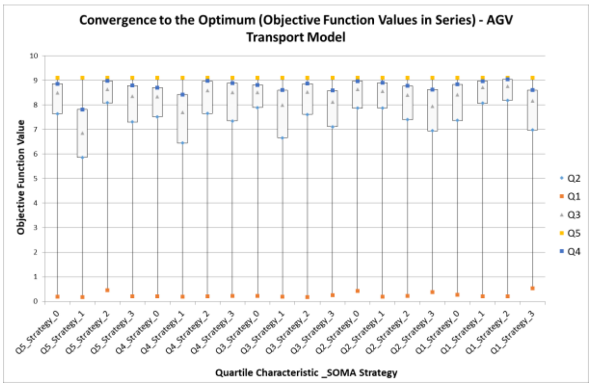

Figure 7: Convergence to the Optimum (Objective Function Values in Series) - Objective Function Maximization

5.6. The Frequency of Using the Database Records

Because we mapped all the possible solutions and their objective function values in the search space and found the lower and upper bounds of the objective function of the simulation model we could count the relative frequencies of the feasible solutions objective function values between the specified range. We divided the whole interval of objective function values into 100 smaller parts - intervals ($n = 100$) with the same step:

$$\varepsilon_F = \frac{\left|F(\hat{\boldsymbol{X}}) - F(\check{\boldsymbol{X}})\right|}{n} \tag{46}$$

where n denotes the number of smaller intervals of the objective function values of feasible solutions with the defined size of the interval - ε_F.

We calculated how often the objective function values occur within different ranges of objective function values:

$$FR_j, j \in [1, n] \tag{47}$$
$$= \begin{cases} FR_j + 1 \ \text{if} \left(F(\boldsymbol{X}_i) \geq \varepsilon_F * (j-1) + F(\check{\boldsymbol{X}})\right) \\ \quad \wedge \left(F(\boldsymbol{X}_i) < F(\check{\boldsymbol{X}}) + \varepsilon_F * (j-1) + \varepsilon_F\right), \\ \quad i \in [1, NCS] \\ FR_j + 0 \quad \text{else} \end{cases}$$

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 the interval of the objective function; NCS denotes the number of feasible solutions in the interval of the objective function value; ε_F denotes the size of the range of the smaller interval.

We calculated the percentage relative frequency of each smaller interval.

Figure 8 shows 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. These data series reflect the quality of the feasible solutions in the search space of the modelled problem.

We also tested the series of the boxplot characteristics of the weighted sum 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 different SOMA strategies using the boxplot characteristics series – see Figure 9.

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 tend towards a strong convergence on the global optimum. If we select less suitable optimization method settings the relative proportion starts to increase in other areas of the interval. Appropriate setting of the optimization method leads to a lower number of optimization experiments to find the global optimum (a steeper convergence on the global optimum and a lower relative frequency in the area of global extremes of the objective function values).

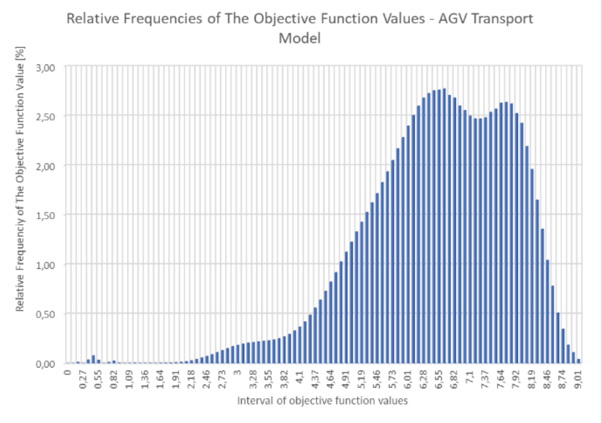

Figure 8: Relative Frequencies of The Objective Function Values – Objective Function Maximization

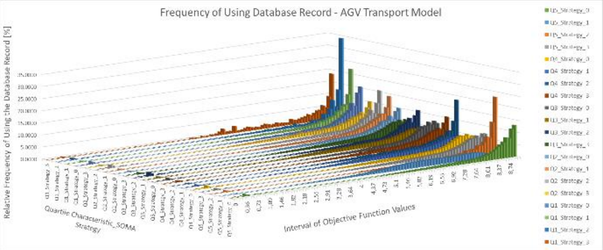

Figure 9: Frequency of Using Database Record - Objective Function Maximization

6. CONCLUSION

The testing of the SOMA optimization method confirmed that it is quite a successful and generic optimization method according to the different objective function landscapes of our tested AGV Transport model. This method found the global optimum or candidate solutions even in the case of the worst settings of this optimization method.

If we focus on the optimization method success and the convergence to the optimum during the optimization process, the AllToAllAdaptive strategy and AllToRand are successful strategies of the SOMA optimization method. These strategies are prone to wrong optimization method parameters settings in our discrete event simulation model. These strategies are followed by the SOMA strategy AllToOne.

The AllToAll strategy is not useful when the second termination allows a small number of simulation experiments to be performed. This strategy supports the exploration and it needs to perform more simulation experiments to find the global optimum We tested this strategy with a higher number of simulation experiments, which SOMA can perform in the series.

ACKNOWLEDGMENTS

This contribution has been prepared within the project LO1502 'Development of the Regional Technological Institute' under the auspices of the National Sustainability Programme I of the Ministry of Education of the Czech Republic aimed at supporting research, experimental development and innovation.

REFERENCES

Bučková M., Krajčovič M., Edl M., 2017. Computer Simulation and Optimization of Transport Distances of Order Picking Processes. Procedia Engineering, 192, 69-74.

Borda, M., 2011. Fundamentals in Information Theory and Coding. Berlin: Springer Berlin Heidelberg.

Elsayed, S. M., Ruhul A. S., Daryl L. E., 2013. An Improved Self-Adaptive Differential Evolution Algorithm for Optimization Problems. IEEE Transactions on Industrial Informatics, 9 (1), 89-99.

Koblasa, F., Vavrousek, M., Manlig, F., 2017. Three-dimensional Bin Packing Problem with heterogeneous batch constraints. In: P. Prazak, ed. Mathematical Methods in Economics, 330–335.

Li Y. L., Zhan Z. H., Gong Y. J., Chen W. N., Zhang J., Li, Y. 2015. Differential Evolution with an Evolution Path: A DEEP Evolutionary Algorithm. IEEE Transactions on Cybernetics, 45(9), 1798–1810.

Pohlheim, H., 2006. GEATbx: Example Functions. Available from: http://www.geatbx.com/docu/fcnindex-01.html#P204_10395 [accessed 20 November 2011]

Raska, P., and Ulrych, Z., 2014. Hierarchical approach to developing a logistic discrete event simulation model using Automated Guided Vehicles. 26th European Modeling and Simulation Symposium, pp. 205-211. Bordeaux (France).

Volna, E., 2012. Evolutionary Algorithms and Neural Networks (in Czech language: Evolucni algoritmy a neuronove site). University of Ostrava. Available from: http://www1.osu.cz/~volna/Evolucni_algoritmy_a_neuronove_site.pdf [accessed 5 March 2016]

Weise, T., 2009. E-Book "Global Optimization Algorithms - Theory and Application" 2nd Edition. Available from: http://www.it-weise.de/projects/book.pdf [accessed 2 February 2011]

Zelinka, I., 2016. SOMA—Self-organizing Migrating Algorithm. In: Studies in Computational Intelligence. Springer Verlag, 3–49.

Zelinka, I., Snasel, V. and Abraham, A., 2013. Handbook of Optimization: From Classical to Modern Approach. Intelligent Systems Reference Library, Berlin: Springer.

Zelinka, I., Vybíral J., 2005. SOMA - Source Codes. Tomas Bata University in Zlín. Available from: http://www.ft.utb.cz/people/zelinka/soma/ [accessed 6 March 2012]

ABOUT THE AUTHORS

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.