Methodology for Evaluating Optimization Experiments

Pavel Raska[1,*] and Zdenek Ulrych[1]

1Department of Industrial Engineering - Faculty of Mechanical Engineering, University of West Bohemia, Univerzitni 22, Pilsen, 306 14, Czech Republic

*Corresponding author. Email address: praska@kpv.zcu.cz

Abstract

The paper presents a methodology for evaluating optimization experiments performed on simulation models (discrete event simulation models and Bin Packing Problem) which focus on various problems in industrial engineering. We created an evaluation application to validate our methodology. The application evaluates the simulation experiments generated by a simulation optimizer. The first goal is to evaluate the behaviour of the tested optimization methods – evaluate a series of optimization experiments (replicated simulation optimization experiments) with different settings of the optimization algorithms according to different characteristics of the optimization (e.g. efficiency of finding a global optimum, quality of a found solution, speed of finding an optimum, etc.). The next goal is to evaluate the difficulty of the simulation optimization - to determine relative frequencies of the number of simulation experiments until the optimum of the objective function is found for different intervals. The objective function represents the quality of a possible solution in the n-dimensional search space for the tested simulation model.

Keywords: Simulation optimization, Heuristic methods, Evaluation, Global optimization, Discrete Event Simulation

1. Introduction

The paper deals with the methodology of evaluation of discrete event simulation optimization experiments involving the use of different optimization methods focused on a global optimization problem. Our department is also focused on the simulation of processes in industrial companies. Many of these simulation studies are NP hard problems where we cannot often evaluate all possible solution candidates (a solution candidate is a possible solution of the modelled problem) in a search space even though the search space is usually boundary-constrained:

$$\tilde{X} = \prod_{j=1}^{n} \tilde{X}_j = \prod_{j=1}^{n} [a_j, b_j], a_j \le b_j, a_j, b_j \in \mathbb{R} \tag{1}$$

where $\tilde{X}$ denotes the search space - the domain of the input parameters of the discrete event simulation model (space containing possible solution of the modelled problem); j denotes the index of the j-th decision variable of the simulation model -the j-th simulation model input parameter; n denotes the dimension of the search space (the number of decision variables); a_j denotes the lower bound of the interval of the j-th decision variable; b_j denotes the upper bound of the interval of the j-th decision variable.

The possible solution to the modelled problem is represented by the settings of the simulation model input parameters and it is defined as follows:

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

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.

Each possible scenario of the modelled problem is evaluated by the objective function – suitability of this

possible solution:

$$F(\mathbf{X}) \forall \mathbf{X} \in \tilde{X} \qquad (3)$$

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

The main goal of these simulation studies is to find the best possible scenario − the optimum (if we didn´t map all the search space, the found solution represents the solution candidate) of the modelled problem.

$$\breve{\mathbf{X}} = \operatorname{argmin}_{\mathbf{X} \in \tilde{X}} F(\mathbf{X}) = \{\breve{\mathbf{X}} \in \tilde{X} : F(\breve{\mathbf{X}}) \leq F(\mathbf{X}) \forall \mathbf{X} \in \tilde{X}\} \qquad (4)$$

where $\breve{\mathbf{X}}$ denotes the global minimum of the objective function (if the objective function is minimized); $F(\breve{\mathbf{X}})$ denotes the objective function value of the minimum − optimum (the function maximization can be converted to function minimization by multiplying the objective function value by -1).

Much of the simulation software used today for modelling company processes (e.g. Arena (Rockwell, 2020), Witness (LANNER, 2020), PlantSimulation (Siemens, 2019), etc.) uses its own integrated external simulation optimizers. (Management, 2020)

These simulation optimizers are black boxes in many cases where a user cannot set up all the parameters of the implemented optimization methods and some simulation optimizers cannot affect all the parameter types of the simulation model. We developed our own simulation optimizer (Client − Server application), a manager for the optimization experiments (remote control of the simulation optimizers) and an evaluator for the optimization experiments.

Certain parts of the algorithms contain randomness (generating a possible solution for the actual solution, selection, mutation, etc.). Due to escaping from the local extremes of the objective function we must replicate optimization experiments several times to assess the optimization method behaviour. 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 our methodology on different discrete event simulation models. The methodology is illustrated on the simple discrete event simulation model of Penalty2. Number two denotes the number of decision variables of the simulation model. This simulation model represents the production of two types of product at a workshop consisting of eight workstations. Each workstation contains a different number of machines. Each product has a specific sequence of manufacturing processes and machining times. We specified the initial fixed solution which represents a bad solution to a modelled problem. This

solution is part of the set of initial solutions generated by the optimization method. The optimization method can randomly generate initial possible solutions near the area of the global optimum, and it can distort the process of searching for the optimum. This specification also allows us to calculate the percentage improvement of the optimization process. 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 total time is the output obtained from the simulation model after the simulation run. Controls of this discrete event simulation model are the arrival times of each product in the system). The objective function is minimized. The global minimum of the objective function equals 100.7093. A detailed description of this model can be found in (Raska & Ulrych, 2015).

We mapped all the possible solutions in the search space of the simulation models and evaluated them by the relevant objective function. We converted the decision variables values of the possible solution located in the search space of the simulation model together with the objective function value of this possible solution into one record stored in the database. Each discrete event simulation model has its own database of evaluated possible solutions. The simulation optimizer downloads the relevant database of the simulation model from the server before launching the optimization process. The simulation optimizer searches for this possible solution in its downloaded database which is saved in the PC´s internal memory. This approach speeds up the evaluation process of the generated possible solution by the optimization method. If the internal database doesn´t contain a possible solution for the problem, the simulation software is launched. After the simulation run the result is evaluated with the objective function. The decision variables values and objective function value are converted to one record (one number) and they are stored in the internal memory. The record is also sent to server. It enables to other launched simulation optimizers download this record to their internal database. The simulation optimizer doesn´t need to perform simulation run with the same setting of decision variables values.

We implemented different stochastic pseudo gradient, metaheuristic, evolutionary and swarm optimization methods to compare their efficiency in the search for the optimum. Selected methods are: Random Search (Weise, 2009), Hill Climbing (Volna, 2012) (Majer, 2003), Tabu Search (Monticelli, Romero, & Asada, 2007), (Weise, 2009) (Volna, 2012), Local Search (Majer, 2003), Downhill Simplex (Clerc, 1999) (Eberhart & Shi, Comparing inertia weights and constriction factors in particle swarm optimization, 2000) (Ma, Zhang, & Xu, 2015) (Tvrdik, Stochastic

Algorithms for Global Optimization (in Czech language: Stochastické algoritmy pro globální optimalizaci), 2010), Simulated Annealing (Weise, 2009) (Tvrdik, Evolutionary algorithms - Study Texts (in Czech language Evoluční algoritmy - učební texty), 2004), Differential Evolution (Volna, 2012) (Tvrdik, Stochastic Algorithms for Global Optimization (in Czech language: Stochastické algoritmy pro globální optimalizaci), 2010), Evolution Strategy (Volna, 2012) (Tvrdik, Stochastic Algorithms for Global Optimization (in Czech language: Stochastické algoritmy pro globální optimalizaci), 2010) (Marik, Stepankova, & Lazansky, 2001), Particle Swarm Optimization (Clerc, 2010) and Genetic algorithm (A.J. & P.D., 2015), (Kumar & Gopal, 2013), (Ribagin, Roeva, & Pencheva, 2016), (Ahmed, 2010). We tested different types of optimization strategies of the optimization methods e.g. the Evolution strategy uses different types of selection: Truncation (Cut-off value parameter), Random, Roulette wheel, Tournament (Number of other contestants per tournament parameter) and Ordered selection. All types of selection were tested with or without the possibility to copy the individual to the population.

The self-organizing migrating algorithm (SOMA) uses different types of strategies of the movement of individuals to a leader - All to One, All to all, All to all - adaptive random, All to one – random. The particle swarm optimization method (PSO) uses different strategies of searching: Inertia weight method (Eberhart & Shi, 2001), (Eberhart & Shi, 2000); Constriction factor (Clerc, 1999), (Eberhart & Shi, 2000); Radius particle swarm optimization (Anantathanavit & Munlin, 2013); Interpersonal learning PSO (Ma, Zhang, & Xu, 2015); A distance-based locally informed PSO (Qu, Suganthan, & Das, 2013); PSO with an aging leader and challengers (Chen et al., 2013). Genetic algorithm uses different types of crossover: One-point, Three parents, Two-point, Uniform, Voting recombination crossover. We also tested the different types of generation strategy of the Genetic algorithm: Performance and Tracking strategy. The tested mutation strategies were: Displacement, Flip bit, Insertion, Partial shuffle, Reverse sequence, Uniform mutation, etc.

2. Optimization experiments

It is necessary to specify the basic attributes of the optimization experiments for each simulation model in order to test the selected optimization methods and to evaluate the effect of setting different parameters of optimization methods to find the optimum or suboptimum. Basic attributes are: Specification of the decision variables in the search space - lower and upper boundary, step on the axis of the decision variable; Search space specification; Specification of the simulation run time of the simulation model; Specification of the objective function representing the goal of the simulation study; Specification of termination criteria; Mapping the objective function

within the search space of the simulation model; Identification of the global optimum of the objective function in the search space.

The same termination criteria were satisfied for each tested series. We specified two termination criteria – Value to Reach (because we mapped all the solution candidates in the search space, and we obtained the objective function values of the global minimum of the search space) and the maximum number of simulation experiments that the optimization method can perform in the optimization experiment in each series. We specified the tolerated deviation from the objective function value of the global optimum for each simulation model:

$$\varepsilon = |F(\mathbf{X}) - F(\mathbf{X}^*)| \tag{5}$$

where $F(\mathbf{X})$ denotes the objective function of the found solution candidate (possible solution to a modelled problem); $F(\mathbf{X}^*)$ denotes the objective function value of the global optimum. If the tolerated deviation (for each simulation model) meets the following criterion, the optimization experiment is stopped:

$$\varepsilon \leq 0.001 \tag{6}$$

We calculated the maximum number of simulation experiments that the optimization method can perform in the optimization experiment by 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 is calculated as follows:

$$\delta = \max\{0, 1 - \beta \cdot log\tilde{X}\}, \delta \in [0,1] \tag{7}$$

where $\tilde{X}$ denotes the number of all possible solutions in the search space; β denotes the coefficient of the search space reduction (we set the coefficient $\beta = 0.05$ according to our initial optimization experiments).

We calculated the maximum number of simulation experiments that the optimization method can perform in each optimization experiment after the calculation of the reduction coefficient:

$$\tilde{X}_H = \lfloor 2^{\delta \cdot log_2\tilde{X}} \rfloor \tag{8}$$

3. Evaluation

We defined different indicators of the efficiency of finding the optimum in the simulation model search space and for the analysis of the behaviour of different optimization algorithms depending on the different settings of the algorithm parameters. These evaluation criteria can be divided into the following basic areas:

1. Quality of the found solution
2. Algorithm speeds of finding a solution
3. Weighted sum of different evaluation criteria considering the quality and speed of the algorithm of finding a solution
4. Other properties of the algorithm of finding a

solution

3.1. Quality of the found solution

The basic principle of global simulation optimization is to find the global optimum in the search space. The quality of the solution found by the optimization algorithm can be assessed by the objective function. The quality of all found solutions within the performed series can be evaluated by the calculation of the quartile characteristics and their visualization using a boxplot chart.

Boxplot charts are generated in the developed application and show the range of values of the objective function from minimum to maximum, as well as the 1st and 3rd quartiles dividing the set of found solutions (solutions are sorted according to the objective function values) into quarters. It is possible to quickly visually evaluate 25% of the found solutions whose values of the objective function are less than or equal to the lower (first) quartile. A similar principle applies to the upper (third) quartile, where 25% of the solutions are located between the upper quartile and the maximum. A median divide the set of values of the objective function into two approximately equal halves. The following figure shows an example of calculated quartile characteristics of the objective function values of the solutions found by different optimization methods considering different tested series - different parameter settings of optimization methods. The next Boxplot chart shows the calculated boxplot characteristics of a Penalty2 discrete event simulation model – see **Figure 1**. The information about the minimization or maximization of the objective function value should be placed in the title of the boxplot chart generated by the evaluation application. It helps the user to orientate faster during the visual evaluation of the series results. Another possible solution for quick orientation about the minimization or the maximization of the objective function is by using different colours in the boxplot, including the information in the legend of the chart – see **Figure 1**. We can see that the Local search provides a poor solution candidate in the case of the Penalty2 simulation model. The boxplot´s median, the upper quartile and the maximum of the objective function value of the found optima are identical, which is not desirable if the objective function is minimized. Other pseudo-gradient methods such as Hill climbing and Tabu search found better solution candidates (very small interquartile range, i.e. the distance between the upper and lower quartile). Another problematic optimization method used for the optimization of the Penalty2 simulation model is the Genetic algorithm. It´s interquartile range is bigger than the interquartile ranges of other optimization methods. The Genetic algorithm needs more generation (or a bigger number of individuals in the population) to explore the search space. The advantage of this method is the ability to get out from the local extremes of the multimodal objective function by using a diversity of individuals

incorporated into the population.

The results of optimization methods searching for the optimum are also significantly affected by the number of optimization method parameters. This number is related to the number of tested series (a larger number of possible method settings). We performed 15 series (different settings of optimization method parameters where each series contains 5 optimization experiments) of Local search method parameter (this method contains one method parameter which is set with different values), 90 series of the Hill Climbing method (two method parameters), and 900 series of the Tabu Search method (four method parameters). If the meaning of the optimization methods parameter is the same in different methods, we set each method with the same step value, lower and upper limits of this parameter.

Many of the optimization methods are very sensitive to setting their parameters hence the application also allows the user to filter 25% of the best series. These series are between the minimum and the 1st quartile in the case of objective function minimization (if the objective function is maximized, 25 % of the best series are between 3rd quartile and maximum). The boxplot characteristics of the objective function values of these series are calculated and displayed in the boxplot chart – see **Figure 2**.

It can be deduced from the chart that the Hill Climbing method is less prone to parameter settings than the Tabu Search method.

A very common problem of global optimization is that the number of solutions (elements in the search space - an acceptable solution for a discrete event simulation model) is so large that it is not possible, or it is very difficult, to test all possible solutions - NP hard problem. Therefore, it is not possible to unambiguously determine whether the solution found by the optimization algorithm is a global optimum or is only a local extreme of the objective function. Due to the absence of such information, it is appropriate to use another performance indicator in the application - calculate the quartile characteristics of the percentage improvement of the objective function value of the possible solution found by the optimization method and the objective function of the fixed initial solution (fixed value of the simulation model input parameters). This information is displayed in the Boxplot chart of the evaluation application – see **Figure 3**.

3.2. Speed of finding a solution by the algorithm

The speed of finding the optimum is usually one of main criteria for evaluating the efficiency of the algorithm. We defined the same termination criteria for each series and the global optimum in the search space is known. We can compare the speed of finding the optimum (the number of simulation model runs to find the global optimum) in each series. The following boxplot chart is generated by the evaluation application

according to calculated quartile characteristics of an algorithm speed of finding a solution. The vertical axis contains the number of simulation experiments to find the best solution candidate in the performed series by the optimization methods. The following boxplot chart shows the calculated boxplot characteristics of the speed of finding the optimum by the algorithm – see **Figure 4**. A solution candidate can also be a suboptimum of the objective function. The evaluation application also generates the boxplot chart of the number of simulation experiments until the suboptimum is found (the objective function value of the solution candidate doesn't belong to the tolerated deviation from the objective function value of the global optimum).

The application also generates a bar chart containing the percentage relative frequencies of the number of performed simulation experiments until the optimum was found. We filtered the optimization experiments where the optimum of the objective function was found:

$$\varepsilon \geq |F(\mathbf{X}_i) - F(\mathbf{X}^*)| \qquad (9)$$

where $F(\mathbf{X}_i)$ denotes the objective function of the found solution candidate of the i-th optimization experiment (the solution whose objective function is in the tolerated deviation from the value of the optimum of the objective function $\varepsilon = 0.001$); $F(\mathbf{X}^*)$ denotes the objective function value of the global optimum.

We divided the interval of the maximum number of simulation experiments that the optimization method can perform in the optimization experiment into smaller intervals (with the same step):

$$\gamma = \frac{\tilde{X}_H}{q} \qquad (10)$$

where q denotes the number of smaller intervals of the number of simulation experiments that the optimization method can perform to find the optimum (we set $q = 100$); γ denotes the defined size of the interval.

We calculated the relative frequencies of the number of simulation experiments that the optimization method performed to find the optimum of the objective function for each specified range:

$FR_j, j \in [1, q]$

$$= \begin{cases} FR_j + 1 \text{ if } \binom{NSE_{OE_i} \geq \gamma *}{(j-1)} \wedge \binom{NSE_{OE_i} <}{\gamma * (j-1) + \gamma}, i \in [1, N_{OE} \cdot N_{s.}} \\ FR_j + 0 \quad \text{else} \end{cases} \quad \begin{matrix}(11\\)\end{matrix}$$

where FR_j denotes the frequency of the j-th small interval; j denotes the index of the small interval of the number of simulation experiments that the optimization method performed, NSE_{OE_i} denotes the number of the simulation experiments that the optimization method performed to find the solution candidate of the i-th optimization experiment; N_{OE} denotes the number of the performed optimization experiments of the series; N_s denotes the number of

series of optimization method; γ denotes the size of the range of the smaller interval. The percentage relative frequency is calculated as follows:

$$PRF_j = \left(\frac{FR_j}{N_{OE} \cdot N_s}\right) * 100[\%], j \in [1, q] \qquad (12)$$

Where PRF_j denotes the percentage relative frequency of the j-th interval; $N_{OE} \cdot N_s$ denotes the number of the performed simulation experiments of all series of the optimization method.

The following figure shows the percentage of the calculated relative frequencies of the number of simulation experiments that the optimization method – Evolution strategy - performed to find the optimum considering the intervals of the number of simulation experiments that the optimization method can perform to find the optimum – see **Figure 5**. If we compare this chart with the chart of the Particle swarm optimization method we can see a big difference between the speed of finding the optimum, although both optimization methods use the basic principle of Darwin´s selection – see **Figure 6**. The evaluation application also provides the analysis of the speed of finding suboptimum.
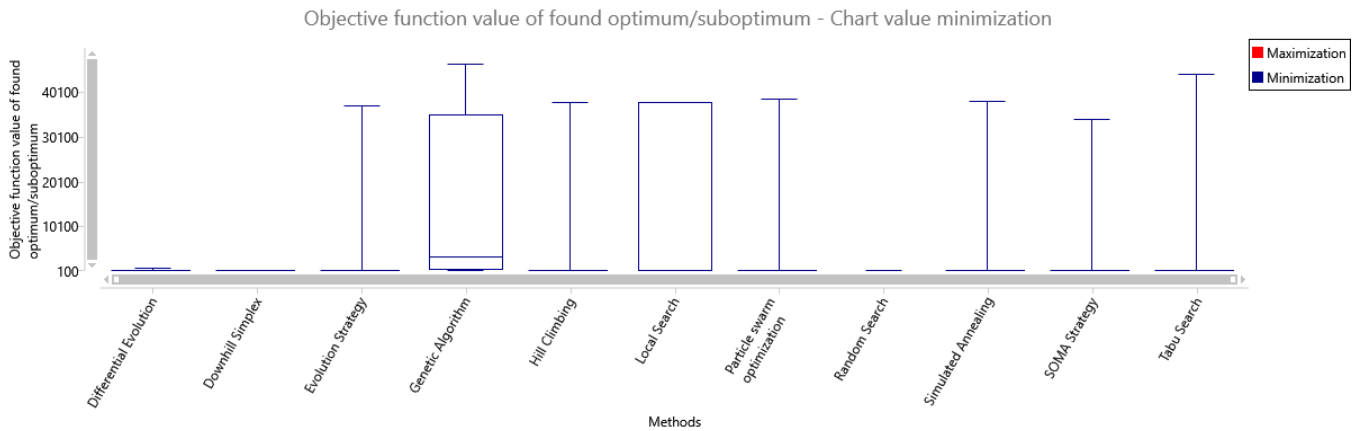
Figure 1. Boxplot chart of the objective function values of solutions found by the optimization methods − All series – Penalty2 simulation model
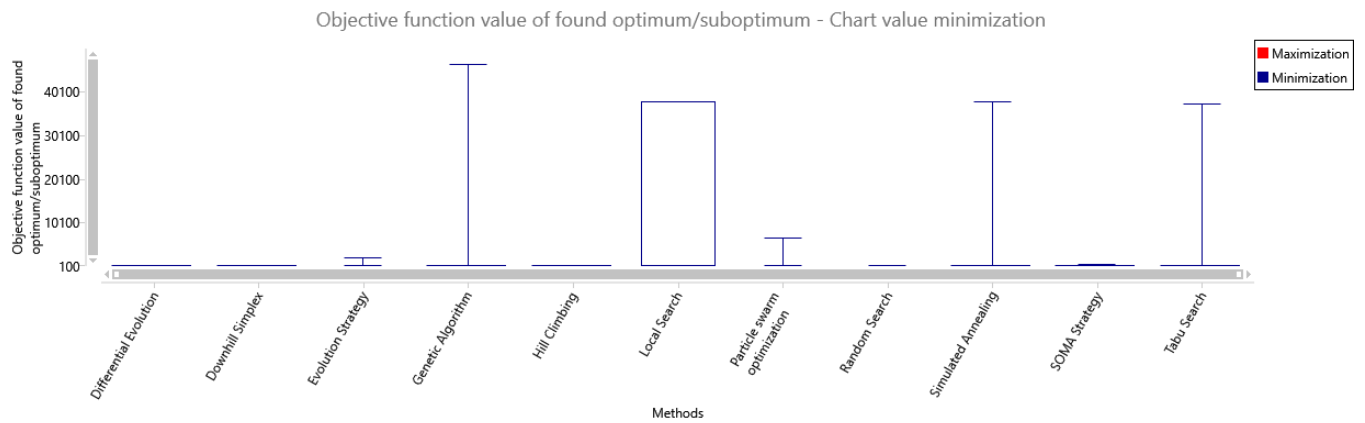

Figure 2. Boxplot chart of the objective function values of found solutions by the optimization methods − 25% of the series– Penalty2 simulation model
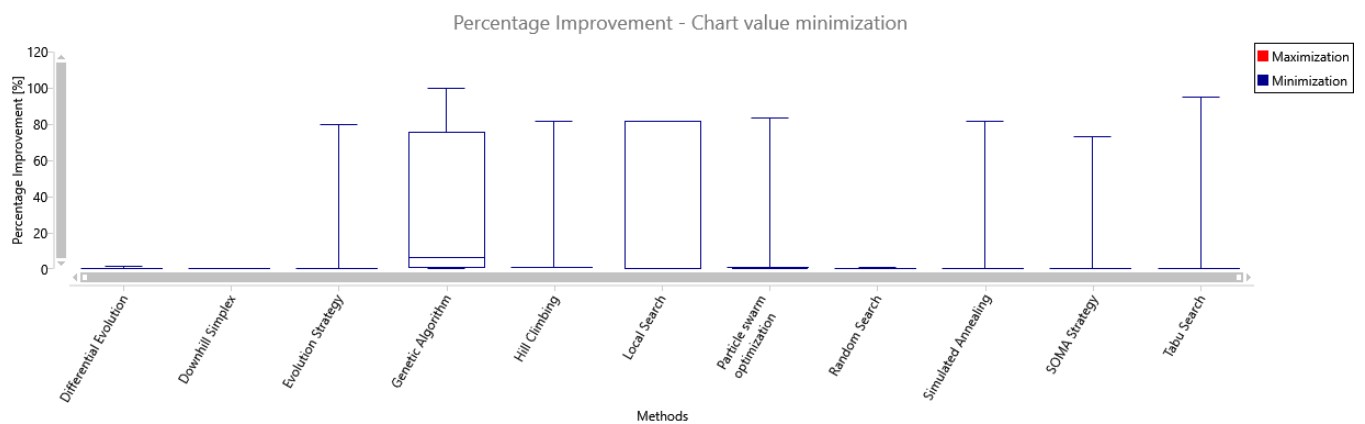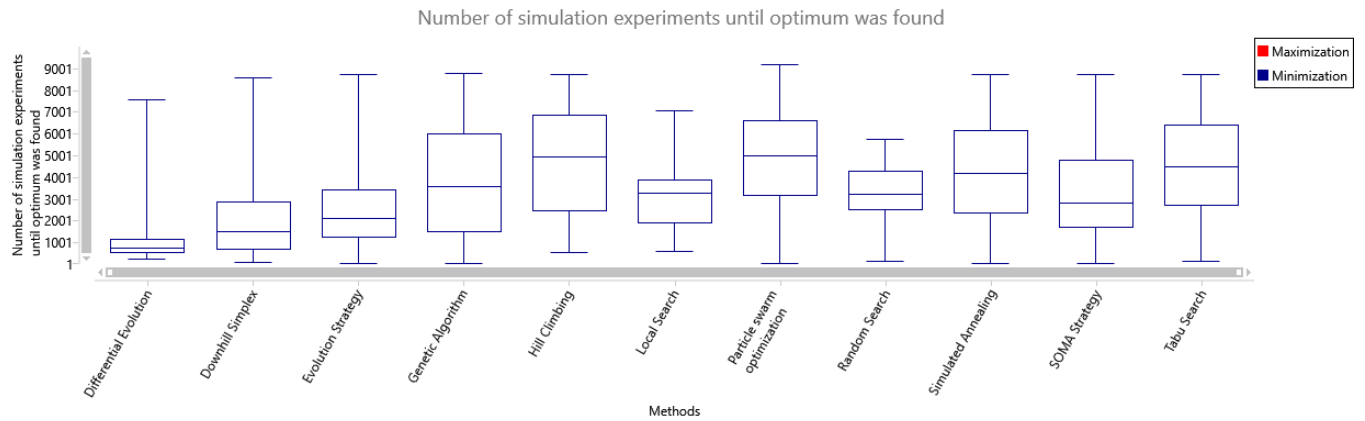

Figure 3. Boxplot chart of the percentage improvement of the objective function values of found solutions by the optimization methods - chart value minimization - Penalty2 simulation model

Figure 4. Boxplot chart of the number of simulation experiments until the optimum was found by the optimization methods – chart value minimization – Penalty2 simulation model
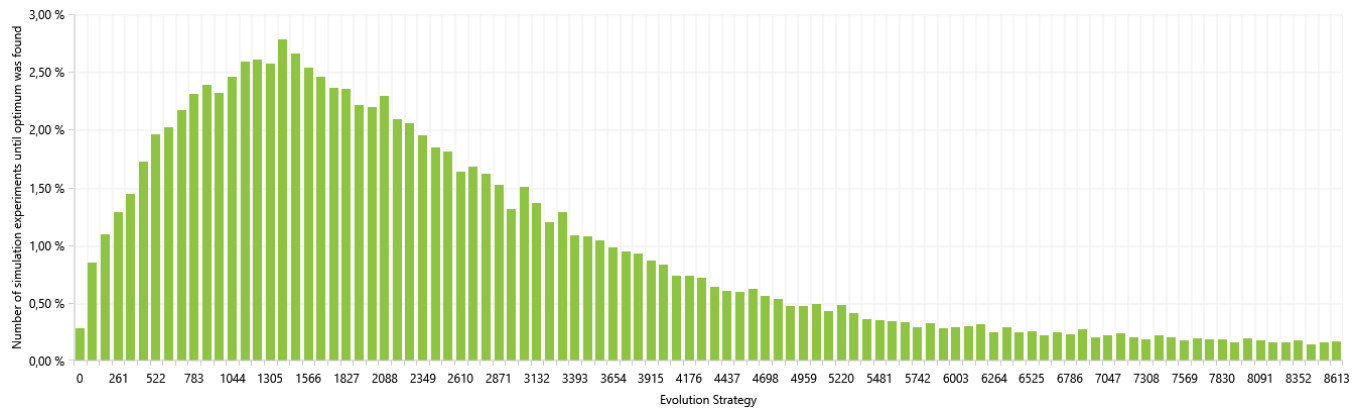
Figure 5. Relative frequencies of the number of simulation experiments until the optimum was found by the Evolution strategy – chart value minimization – Penalty2 simulation model
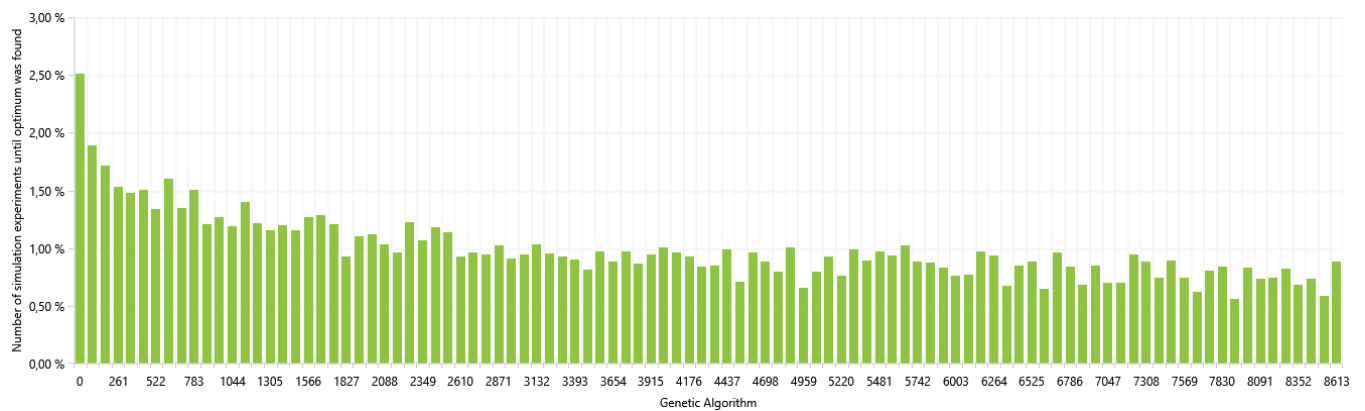
Figure 6. Relative frequencies of the number of simulation experiments until the optimum was found by the Genetic algorithm – chart value minimization – Penalty2 simulation model
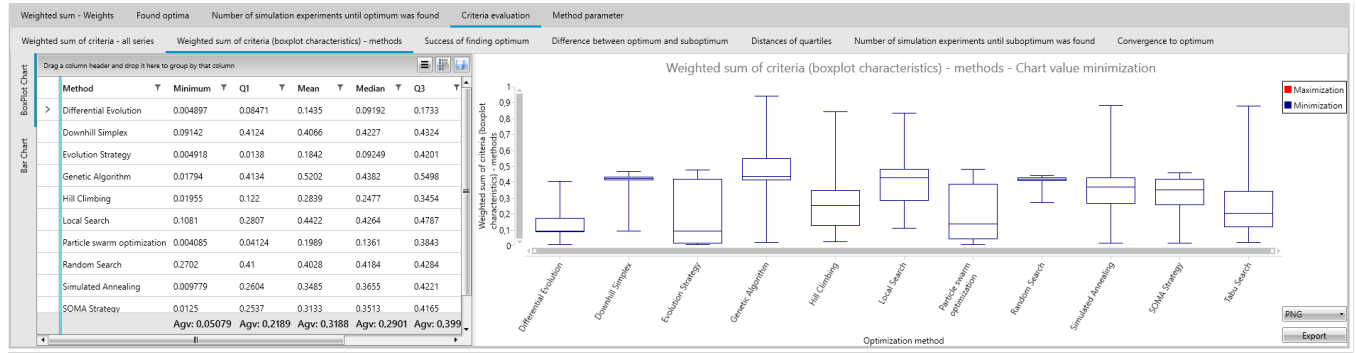
Figure 7. Evaluation criteria − weighted sum − chart value minimization - Penalty2 simulation model

3.3. Weighted sum of different evaluation criteria

The generated boxplot charts provide a quick visual overview of the quality of the solution found and the number of simulation experiments to find the optimal solution. These charts are insufficient if the user needs a comprehensive view. The user also prefers one criterion over another. We defined five basic criteria for evaluating the optimization methods:

1. The success of finding the optimum
2. The difference between the optimum and sub-optimum
3. The distances of quartiles
4. The number of simulation experiments until the suboptimum was found
5. The convergence to the optimum

Criteria are calculated for each series. Each criterion is normalized to a value in the range [0,1]. It is an attempt to minimize the criteria (0 represents the best evaluation of the criterion):

$$f_j \in [0,1] \forall j: j = \{1,2,\dots,5\} \tag{13}$$

where f_j denotes the j-th criterion; j denotes the index of the criterion.

The characteristics of the boxplot charts are calculated according the values of each criterion. These characteristics are displayed in the boxplot chart for each criterion.

3.3.1. *Optimization Algorithm 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:

$$F_{\bar{X}} = \left| F(\breve{\mathbf{X}}) - F(\hat{\mathbf{X}}) \right| \tag{14}$$

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):

$$\Delta F_{\bar{X}} = F(\mathbf{X}^*) - F(\mathbf{X}_{Worst}) \tag{15}$$

where $F(\mathbf{X}^*)$ denotes the objective function value of the found best candidate solution of the search space in all series (global optimum if we mapped all the possible solutions of the search space); $F(\mathbf{X}_{Worst})$ 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 algorithm − value minimization. This criterion is expressed by Pseudo Pascal code − see **Figure 8**.

```
begin
    n_Succ ← 0;
    (*browse all the items of the list*)
    for i ← 0 to Length(X*) − 1 do
        (*optimum or acceptable candidate solution
was found*)
        if |F(X*[i]) − F(X*)| ≤ ε then
            n_Succ ← n_Succ + 1;
        (*standardization - % share of unsuccessful
series*)
        result ← (Length(X*)−n_Succ)/Length(X*);
end;
```

Figure 8. Pseudo Pascal algorithm of the first criterion calculated for one series − finding the global optimum or suboptimum

Where i denotes the index of series; 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 algorithm parameters (number of optimization experiments in one series); $\mathbf{X}^*$ denotes the found best candidate solution (global optimum 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.

3.3.2. The Difference between Optimum and Local Extreme

The second criterion is useful when there is no series which contains any optimum or a suboptimum whose objective function value is within the tolerance of the optimum objective function value (the first criterion equals zero in this case). 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 the objective function value is sorted in ascending order. This sorting can be done using the comparator function, which compares the quality (objective function values) of two possible solutions. If we maximize the objective function:

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

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

After this, the value of the second criterion of the i-th series is calculated using the formula:

$$f_{2\,i} = \left(\frac{F(\mathbf{X}^*) - F(\mathbf{X}^*_i)}{\Delta F_{\bar{X}}} \right) \forall i: i = \{1,2,\dots,s\}, f_{2\,i} \in [0,1] \quad (17)$$

where $F(\mathbf{X}^*)$ denotes the objective function value of the best found candidate solution of the search space in all series; $F(X^*_i)$ denotes the objective function value of the best solution candidate found in i-th series; $\Delta F_{\bar{X}}$ denotes the difference between the objective function value of the found best and worst candidate solutions of the search space in all series; s denotes the number of performed series (different settings of the optimization algorithm parameters).

3.3.3. The Distances of Quartiles

The third criterion expresses the distance between the quartiles of a concrete series. 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 third criterion when the objective function is minimized can be formulated as follows:

$$f_{3\,i} = \frac{f_{3w1\,i} + f_{3w2_i} + f_{3w3\,i} + f_{3w4\,i} + f_{3w5\,i}}{\Delta F_{\bar{X}}}, \forall i: i = \{1,2,\dots,s\}, f_{3\,i} \in [0,1] \quad (18)$$

$$f_{3w5_i} = |F(\mathbf{X}^*_i) - F(\mathbf{X}^*)| \quad (19)$$

$$f_{3w4_i} = w_{4_{f_3}}|F(\mathbf{X}^*_i) - Q_{1_i}| \quad (20)$$

$$f_{3w3\,i} = w_{3_{f_3}}|Q_{1_i} - Q_{2_i}| \quad (21)$$

$$f_{3w2\,i} = w_{2_{f_3}}|Q_{2_i} - Q_{3_i}| \quad (22)$$

$$f_{3w1\,i} = w_{1_{f_3}}|Q_{3_i} - F(\mathbf{X}_{Worst_i})| \quad (23)$$

Where i denotes the index of series; $F(\mathbf{X}^*_i)$ denotes the objective function value of the best solution candidate of i-th series (found minimum of the objective function of the series; $F(\mathbf{X}^*)$ denotes the objective function value of the found best candidate solution of the search space in all series; $w_{4_{f_3}}$ denotes the weight (penalty) of objective function values between the best solution candidate $F(\mathbf{X}^*_i)$ and lower quartile Q_{1_i} of i-th series; $w_{3_{f_3}}$ denotes the weight of objective function values between lower quartile Q_{1_i} and median Q_{2_i} of i-th series; $w_{2_{f_3}}$ denotes the weight of objective function values between median Q_{2_i} and upper quartile Q_{3_i} of i-th series; $w_{1_{f_3}}$ denotes the weight of objective function values between upper quartile Q_{3_i} and the objective function value of the worst found possible solution $F(\mathbf{X}_{Worst_i})$ of i-th series; s denotes the number of performed series (different settings of the optimization algorithm parameters); $\Delta F_{\bar{X}}$ denotes the difference between the objective function value of the found best and worst candidate solution of the search space in all series;.

$$F_{\bar{X}} = |F(\mathbf{X}) - F(\mathbf{X})| \quad (16)$$

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

$$f_{3w5_i} = |F(\mathbf{X}^*) - F(\mathbf{X}^*_i)| \quad (24)$$

$$f_{3w4_i} = w_{4_{f_3}}|Q_{3_i} - F(\mathbf{X}^*_i)| \quad (25)$$

$$f_{3w3\,i} = w_{3_{f_3}}|Q_{2_i} - Q_{3_i}| \quad (26)$$

$$f_{3w2\,i} = w_{2_{f_3}}|Q_{1_i} - Q_{2_i}| \quad (27)$$

$$f_{3w1\,i} = w_{1_{f_3}}|F(\mathbf{X}_{Worst_i}) - Q_{3_i}| \quad (28)$$

The values of the weights are defined based on the results of the simulation experiments. The user can define the weight value of the length between boxplot characteristics. The following table shows the value specified for each weight for the third, fourth and the fifth criterion − see **Table 1**. The sum of the weights equals one.

Table 1. Specifications of weights for the third, fourth and the fifth criterion

Weight	Value
$w_{1_{f_j}}$	0.05
$w_{2_{f_j}}$	0.1
$w_{3_{f_j}}$	0.25
$w_{4_{f_j}}$	0.6

3.3.4. *The Number of Simulation Experiments Until the Optimum Was Found*

The fourth criterion evaluates the number of performed simulation experiments until the best solution candidate was found in each series. The task is to minimize this evaluation function and it can be formulated as follows:

$$f_{4\,i} = \frac{f_{4w1\,i} + f_{4w2_i} + f_{4w3\,i} + f_{4w4\,i} + f_{4w5\,i}}{\tilde{X}_H}, \forall i: i = \{1,2,\dots,s\}, f_{4_i} \in [0,1] \tag{29}$$

$$f_{4w5_i} = |min_{SE_i} - 1| \tag{30}$$

$$f_{4w4_i} = w_{4_{f_4}}|min_{SE_i} - Q_{1_i}| \tag{31}$$

$$f_{4w3\,i} = w_{3_{f_4}}|Q_{1_i} - Q_{2_i}| \tag{32}$$

$$f_{4w2\,i} = w_{2_{f_4}}|Q_{2_i} - Q_{3_i}| \tag{33}$$

$$f_{4w1\,i} = w_{1_{f_4}}|Q_{3_i} - max_{SE}| \tag{34}$$

Where i denotes the index of the series; min_{SE} denotes the minimum number of simulation experiments that the optimization method performed in the optimization experiment to find the best solution candidate of the i-th series (following boxplot characteristics are related to the number of simulation experiments that the optimization method performed in the optimization experiment to find the best solution candidate of the i-th series); max_{SE} denotes the maximum number of simulation experiments that the optimization method performed in the optimization experiment to find the best solution candidate of i-th series; $w_{4_{f_4}}$ denotes the weight (penalty) of values between min_{SE} and lower quartile Q_{1_i} of the i-th series; $w_{3_{f_4}}$ denotes the weight of values between lower quartile Q_{1_i} and median Q_{2_i} of the i-th series; $w_{2_{f_4}}$ denotes the weight of values between median Q_{2_i} and upper quartile Q_{3_i} of the i-th series; $w_{1_{f_4}}$ denotes the weight of values between upper quartile Q_{3_i} and the max_{SE} of the i-th series; s denotes the number of performed series (different settings of the optimization algorithm parameters); $\tilde{X}_H$ denotes the maximum number of simulation experiments (simulation runs) that the optimization method can perform in each optimization experiment in all series – termination criterion.

3.3.5. *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 algorithm in the optimization experiments in the series. The task is to minimize this evaluation function. The fifth criterion when the objective function is minimized can be formulated as follows:

$$f_{5\,i} = \frac{f_{5w1\,i} + f_{5w2_i} + f_{5w3\,i} + f_{5w4\,i} + f_{5w5\,i}}{\Delta F_{\tilde{X}}}, \forall i: i = \{1,2,\dots,s\}, f_{5\,i} \in [0,1] \tag{35}$$

$$f_{5w5_i} = |F(\mathbf{X}^*_i) - F(\mathbf{X}^*)| \tag{36}$$

$$f_{5w4_i} = w_{4_{f_5}}|F(\mathbf{X}^*_i) - Q_{1_i}| \tag{37}$$

$$f_{5w3\,i} = w_{3_{f_5}}|Q_{1_i} - Q_{2_i}| \tag{38}$$

$$f_{5w2\,i} = w_{2_{f_5}}|Q_{2_i} - Q_{3_i}| \tag{39}$$

$$f_{5w1\,i} = w_{1_{f_5}}|Q_{3_i} - F(\mathbf{X}_{Worst_i})| \tag{40}$$

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

$$f_{5w5_i} = |F(\mathbf{X}^*) - F(\mathbf{X}^*_i)| \tag{41}$$

$$f_{5w4\,i} = w_{4_{f_5}}|Q_{3_i} - F(\mathbf{X}^*_i)| \tag{42}$$

$$f_{5w3\,i} = w_{3_{f_5}}|Q_{2_i} - Q_{3_i}| \tag{43}$$

$$f_{5w2_i} = w_{2_{f_5}}|Q_{1_i} - Q_{2_i}| \tag{44}$$

$$f_{5w2_i} = w_{2_{f_5}}|Q_{1_i} - Q_{2_i}| \tag{45}$$

3.3.6. *Weighted Sum*

The calculated criteria help to find a suitable setting of the parameters of the optimization algorithm. The user can prioritize some criteria over others by setting the weights of individual criteria in the evaluation application. The sum of the weights equals one:

$$w_{f_j} \in [0,1] \forall j: j = \{1,2,\dots,5\}, \sum_{j=1}^{5} w_{f_j} = 1 \tag{46}$$

where w_{f_j} denotes the weight of the j-th criterion.

The following table shows the value specified for each weight for each criterion – see **Table 2**.

Table 2. Specifications of weights for the criteria

Weight	Value
w_{f_1}	0.4
w_{f_2}	0.4
w_{f_3}	0.1
w_{f_4}	0.17
w_{f_5}	0.03

The function using the weighted sum of the criteria multiplied by the weights for the i-th series (minimization of the value representing the negative aspects of the algorithm's behaviour):

$$f_i = w_{f_1} \cdot (1 - f_{1\,i}) + \sum_{i=2}^{5} f_{j\,i} \cdot w_{f_j} \ \forall i: i = \{1,2,\dots,s\} \tag{47}$$

where f_i denotes the weighted sum of specified criteria of the i-th series (criterion minimization), i denotes the index of one series; w_{f_1} denotes the weight of the

first criterion; f_{1_i} denotes the standardised scalar value of the i-th series (conversion of algorithm success to the optimization algorithm failure − minimization of the criterion); f_{j_i} denotes the standardised scalar value of the j-th criterion of i-th series; w_{f_i} denotes the weight of the j-th criterion; s denotes the number of performed series (different settings of the optimization algorithm parameters).

The evaluation application calculates the quality of each series which comprises all the proposed evaluation criteria using the weighted sum as the main criterion. The evaluation application calculates the box plot characteristics - minimum, the first quartile, the median, the third quartile and the maximum - of the whole interval of the weighted sum − see **Figure 7**.

4. Conclusions

We proposed a methodology for evaluating optimization experiments performed on discrete event simulation models focusing on various problems in industrial engineering. The methodology was validated using the evaluation application. The application evaluates the optimization experiments obtained from a different simulation optimizer using Client-server architecture. The application evaluates the behaviour of the tested optimization methods. We tested different settings of the optimization methods parameters. We proposed different evaluation criteria - efficiency of finding a global optimum, quality of a found solution, speed of finding an optimum, relative frequencies of the number of simulation experiments until the optimum of the objective function was found for different intervals etc.

These approaches for evaluating optimization experiments can also be used for the evaluation of the Bin Packing Problem. We used standard optimization methods for solving BPP (e.g. Next fit, Best fit, Shelf algorithm, etc.) together with the metaheuristic methods mentioned in this paper. We can evaluate the efficiency of the implemented method using the evaluation application.

We are currently developing a method for comparing multiple models in the evaluation application.

Acknowledgements

References

A.J., U., & P.D., S. (2015). Crossover Operators in Genetic Algorithms: a Review. *ICTACT Journal on Soft Computing*, *06*(01), 1083−1092.

https://doi.org/10.21917/ijsc.2015.0150

Ahmed, Z. H. (2010). Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator. *International Journal of Biometrics & Bioinformatics (IJBB)*.

Anantathanavit, M., & Munlin, M. A. (2013). Radius particle swarm optimization. In *2013 International Computer Science and Engineering Conference, ICSEC 2013*. https://doi.org/10.1109/ICSEC.2013.6694765

Borda, M. (2011). *Fundamentals in Information Theory and Coding. Fundamentals in Information Theory and Coding.* https://doi.org/10.1007/978-3-642-20347-3

Chen, W. N., Zhang, J., Lin, Y., Chen, N., Zhan, Z. H., Chung, H. S. H., ... Shi, Y. H. (2013). Particle swarm optimization with an aging leader and challengers. *IEEE Transactions on Evolutionary Computation*. https://doi.org/10.1109/TEVC.2011.2173577

Clerc, M. (1999). The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*. https://doi.org/10.1109/CEC.1999.785513

Clerc, M. (2010). *Particle Swarm Optimization. Particle Swarm Optimization.* https://doi.org/10.1002/9780470612163

Eberhart, R. C., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation, CEC 2000*. https://doi.org/10.1109/CEC.2000.870279

Eberhart, R. C., & Shi, Y. (2001). Particle swarm optimization: Developments, applications and resources. In *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*. https://doi.org/10.1109/cec.2001.934374

Kumar, R. R., & Gopal, G. (2013). Novel Crossover Operator for Genetic Algorithm for Permutation Problems. *International Journal of Soft Computing and Engineering (IJSCE)*.

LANNER. (2020). *WITNESS Horizon*. Retrieved June 5, 2020, from Lanner: https://www.lanner.com/en-us/technology/witness-simulation-software.html

Ma, J., Zhang, J., & Xu, L. (2015). Interpersonal Learning Particle Swarm Optimizer. In *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings*. https://doi.org/10.1109/CEC.2015.7256886

Majer, P. (2003). Modern Methods of Production Scheduling In Czech language: Moderní metody rozvrzhování výroby), PhD. Thesis. Brno: Brno - University of Technology, Faculty of Information Technology.

Management, I. -T. (2020). *Simulation Software Survey.* (INFORMS - The Institute for Operations Research and the Management Sciences) Retrieved May 5, 2018, from OR/MS Today Software Surveys: https://pubsonline.informs.org/magazine/orms-today/2019-simulation-software-survey#vendors

Marik, V., Stepankova, O., & Lazansky, J. (2001). *Artificial Intelligence (3)* (Vol. III). (A. Badura, Ed.) Prague: Academia Praha.

Monticelli, A. J., Romero, R., & Asada, E. N. (2007). Fundamentals of Tabu Search. In *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems.* https://doi.org/10.1002/9780470225868.ch6

Qu, B. Y., Suganthan, P. N., & Das, S. (2013). A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation.* https://doi.org/10.1109/TEVC.2012.2203138

Raska, P., & Ulrych, Z. (2015). Comparison of optimisation methods tested on testing functions and discrete event simulation models. *International Journal of Simulation and Process Modelling.* https://doi.org/10.1504/ijspm.2015.071380

Ribagin, S., Roeva, O., & Pencheva, T. (2016). Generalized Net model of asymptomatic osteoporosis diagnosing. In *2016 IEEE 8th International Conference on Intelligent Systems, IS 2016 - Proceedings.* https://doi.org/10.1109/IS.2016.7737488

Rockwell, A. (2020). *Arena Simulation Software.* Retrieved June 5, 2020, from Rockwell Automation: https://www.arenasimulation.com/

Siemens. (2019). *Plant Simulation.* (Siemens) Retrieved June 10, 2020, from Siemens PLM Software: https://www.plm.automation.siemens.com/store/en-us/trial/plant-simulation.html

Tvrdik, J. (2004). *Evolutionary algorithms - Study Texts (in Czech language Evoluční algoritmy - učební texty).* Retrieved February 6, 2012, from Virtual Information Centre for PhD. Students: http://prf.osu.cz/doktorske_studium/dokumenty/Evolutionary_Algorithms.pdf

Tvrdik, J. (2010). *Stochastic Algorithms for Global Optimization (in Czech language: Stochastické algoritmy pro globální optimalizaci).* Retrieved January 05, 2014, from Algorithms of global optimization and their applications: http://www1.osu.cz/~tvrdik/wp-content/uploads/STAGO_10.pdf

Volna, E. (2012). *Evolutionary Algorithms and Neural Networks (in Czech language: Evolucni algoritmy a neuronove site).* Retrieved 5 3, 2016, from http://www1.osu.cz/~volna/Evolucni_algoritmy_a_neuronove_site.pdf

Weise, T. (2009). Global Optimization Algorithms−Theory and Application. *URL: Http://Www. It-Weise. de, Abrufdatum.* https://doi.org/doi=10.1.1.64.8184