Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 69 (2014) 768 – 777

Procedia Engineering

www.elsevier.com/locate/procedia

24th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2013

Testing Optimization Methods on Discrete Event Simulation Models and Testing Functions

Pavel Raska*, Zdenek Ulrych

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

Abstract

The paper deals with testing of selected heuristic optimization methods and their evaluation. We have proposed different techniques which express the success of the optimization method in different ways (the method success, the difference between optimum and local extreme, the distances of quartiles, the number of simulation experiments until the optimum was found). These evaluation techniques use box plot characteristics calculated from the repeated optimization experiments.

Selection and peer-review under responsibility of DAAAM International Vienna

Keywords: Simulation Optimization; Evaluation; Heuristic Optimization Methods; Discrete Event Simulation Models; Testing Function

1. Introduction

Many industrial companies are solving the problem of how to design their system (production system, logistic, etc.) as effectively as possible. We have to say that this problem is affected by many internal or external company factors. There exist many possible scenarios how to solve these problems, but which of these scenarios is the best (optimal)? Is it possible to imagine how a change in the subsystem affects the entire system? We have to say that many of these NP-hard problems are impossible to solve using only the human factor or by static calculation. One of the possible answers to the previous question is the use of discrete event simulation in connection with optimization. Many present simulation software packages use their own integrated simulation optimizers which are black-boxes. We can list some problems of these integrated simulation optimizers: e.g. the user cannot set or tune the parameters

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

Selection and peer-review under responsibility of DAAAM International Vienna
doi:10.1016/j.proeng.2014.03.053

of the optimization method even in the case where the user has appropriate information about the objective function type. The simulation optimizer switches optimization methods (e.g. neural network) to find an effective method and the user does not know which methods were selected and the success of the method; the user cannot implement or modify a possible effective optimization method; no possibility to save experimental data from the simulation to a database (knowledge database generation), etc.

We have developed our own simulation optimizer (automatically varies the simulation models' input parameters values to achieve the best configuration of the simulation models) considering the needs of the department to solve practical simulation projects [1], [2] and the module for testing the implemented optimization methods and the optimization methods parameters settings (this module does the same as the first module but it is mainly focused on evaluation of optimization method behavior). We have proposed and implemented some evaluation techniques to better understand the behavior of the optimization methods.

2. Developed Application – Implemented Optimization Methods

Many integrated simulation optimizers use similar optimization methods e. g. Simulated Annealing, Tabu Search, etc. After the literature review we selected commonly used optimization methods to compare their efficiency in searching for the global optimum in the Search space. These methods are:

- Random Search (a new candidate solution is generated in the search space with uniform distribution - Monte Carlo method)
- Stochastic Hill Climbing (candidate solutions – individuals in the population - are generated in the neighborhood of the best candidate solution from the previous population. Generating new possible solutions is performed by mutation) [3]
- Stochastic Tabu Search (if a new candidate solution is generated, it becomes an element of the Tabu List. This solution cannot be visited again if the aspiration criterion is not satisfied. The method uses the FIFO method of removing the candidate solution from the Tabu List. The user can set whether the new candidate solution is generated using mutation of the best candidate solution from the previous population or the new solution is generated using mutation of the best found candidate solution) [4], [5]
- Stochastic Local Search (a candidate solution is generated in the neighborhood of the best candidate solution.) [3]
- Stochastic Simulated Annealing (a candidate solution is generated in the neighborhood of the candidate solution known from the previous iteration. This generating could be performed through the mutation of a randomly selected gene or through the mutation of all genes. Acceptance of the worse candidate solution depends on the temperature. Temperature is reduced if the random number is smaller than the acceptance probability or the temperature is reduced if and only if a worse candidate solution is generated. If the temperature falls below the specified minimum temperature, temperature is set to the initial temperature) [4], [5]

We implemented the basic principle of evolutionary algorithms into some of these optimization methods (generating a whole population instead of one possible solution in order to avoid getting stuck on a local optimum). Previous testing of optimization methods confirmed that generating one solution leads to premature convergence (depending on objective function type). We united different variants of selected optimization methods. The user can choose a different variant of the optimization method by clicking on the checkbox.

We have also tested other optimization methods used in optimization of continuous simulation:

- Downhill Simplex (this heuristic method uses a set of n + 1 linearly independent candidate solutions - n denotes search space dimension - Simplex. The method uses four basic phases – Reflection, Expansion, Contraction and Reduction) [5], [6]
- Differential Evolution (the selection is carried out between the parent and its offspring (the offspring is created through a crossover between the parent and the new individual which was created through the mutation of four selected individuals and the best one selected from the population – BEST method. The optimization method uses General Evolution and the Ali and Törn adaptive rule) [6], [7], [8], [9]

- Evolution Strategy (the optimization method uses Steady State Evolution – population consists of children and parents with good fitness. The individual - child - is generated in the neighborhood of the other individual – parent. The method uses the Rechenberg 1/5th-rule. The population is sorted according to the objective values - Rank-Based Fitness Assignment. The optimization method uses Tournament selection) [6], [10], [11]

3. Discrete event simulation models and testing functions

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

- The Manufacturing System and Logistics model - this discrete event simulation model represents the production of different types of car lights in a whole production system. The complex simulation model describes many processes; for example, logistics in three warehouses, production lines, 28 assembly lines, painting, etc. The objective function is affected by the sum of the average utilization of all assembly lines and average transport utilization. The objective function is maximized. Controls are the number of forklifts responsible for: transport of small parts from the warehouse to the production lines and assembly lines, transport of large parts from the warehouse to the assembly lines, and the transport of the final product from the assembly lines to the warehouse.
- The Penalty model - this simulation model represents a production line which consists of eight workstations. Each workstation contains a different number of machines. Each product has a specific sequence of manufacturing processes and machining times. The product is penalized if the product exceeds the specified production time. A penalty also occurs if the production time value is smaller than the specified constant (this rule is defined because premature production leads to increasing storage costs – the JIT product). The objective function is affected by the total time spent by the product in the manufacturing system. The objective function is minimized. Controls of the production line simulation model are the arrival times of each product in the system.
- The Assembly Line model - this model represents an assembly line. Products are conveyed by conveyor belt. The assembly line consists of eleven assembly workplaces. Six of these workplaces have their own machine operator. The rest of the workplaces are automated. A specific scrap rate is defined for each workplace. At the end of the production line is a sorting process for defective products. The objective function reflects the penalty which is affected by the number of defective products and the palettes in the system. The objective function is maximized. The input simulation model parameters (controls) are the number of fixtures in the system and the number of fixtures when the operator has to move from the first workplace to the eleventh workplace to assemble waiting parts on the conveyor belt.

We tested the implemented optimization methods on four standard testing functions. All testing functions were minimized.

- De Jong´s function – the function is a continuous, convex and unimodal testing function. The function definition:

$$F(\mathbf{X}) = \sum_{j=1}^{n} x_j^2 \tag{1}$$

where $F(\mathbf{X})$ denotes the objective function; j denotes index of control; n denotes the dimension of the search space; x_j denotes the value of control.
- Rosenbrock´s function - Rosenbrock´s (Rosenbrock's valley, Rosenbrock's banana) function is a continuous, unimodal and non convex testing function. The function definition:

$$F(\mathbf{X}) = \sum_{j=1}^{n-1} 100 \cdot (x_j^2 - x_{j+1})^2 + (1 - x_j)^2 \tag{2}$$

- Michalewicz´s function - Michalewicz´s function is a multimodal test function (*n*! local optima). The parameter m defines the "steepness" of the valleys or edges. Larger *m* leads to a more difficult search. For very large *m* the function behaves like a needle in a haystack (the function values for points in the space outside the narrow peaks give very little information on the location of the global optimum). The function definition: [12]

$$F(\mathbf{X}) = -\sum_{j=1}^{n} \sin(x_j) \cdot \left(\sin\left(\frac{j \cdot x_j^2}{\pi} \right) \right)^{2 \cdot m} \tag{3}$$

$$j = 1 : n, 0 \le x_j \le \pi \tag{4}$$

We selected $m = 5$ in our simulation model.

- Ackley´s functions - Ackley´s function is a multimodal test function. This function is a widely used testing function for premature convergence. The function definition:

$$F(\mathbf{X}) = -20 \cdot \exp\left(-0.02 \cdot \sqrt{\frac{1}{n} \cdot \sum_{j=1}^{n} x_j^2} \right) - \exp\left(\frac{1}{n} \cdot \sum_{j=1}^{n} \cos 2 \cdot \pi \cdot x_j \right) + 20 + \exp(1) \tag{5}$$

$$j = 1 : n, -30 \le x_j \le 30 \tag{6}$$

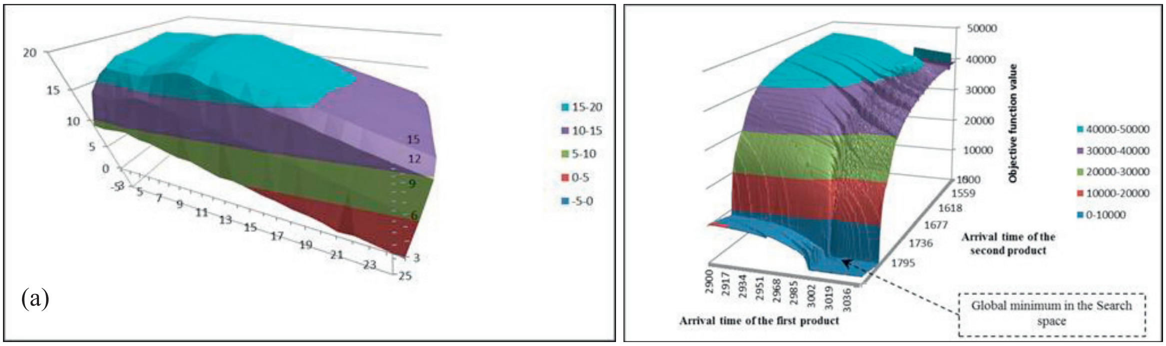

Fig. 1. (a) Objective Function - The Manufacturing System and Logistics Discrete Event Simulation Model - Number of Forklifts for Large Parts = 14; (b) Objective Function – The Penalty Discrete Event Simulation Model.

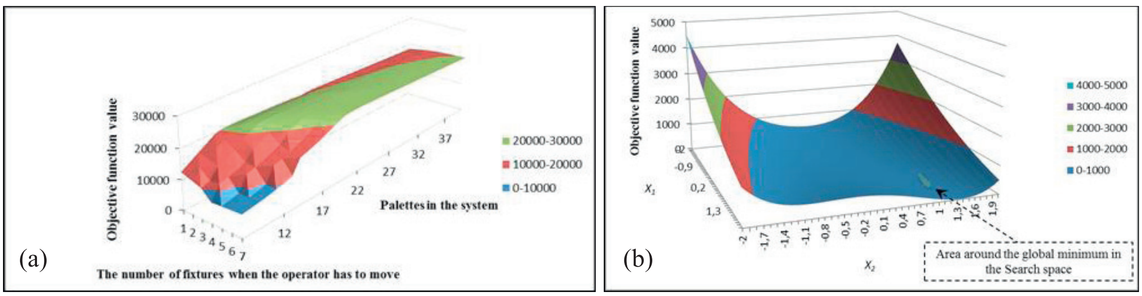

Fig. 2. (a) Objective function - The Assembly Line (discrete event simulation model); (b) Objective Function - Rosenbrock´s Function.

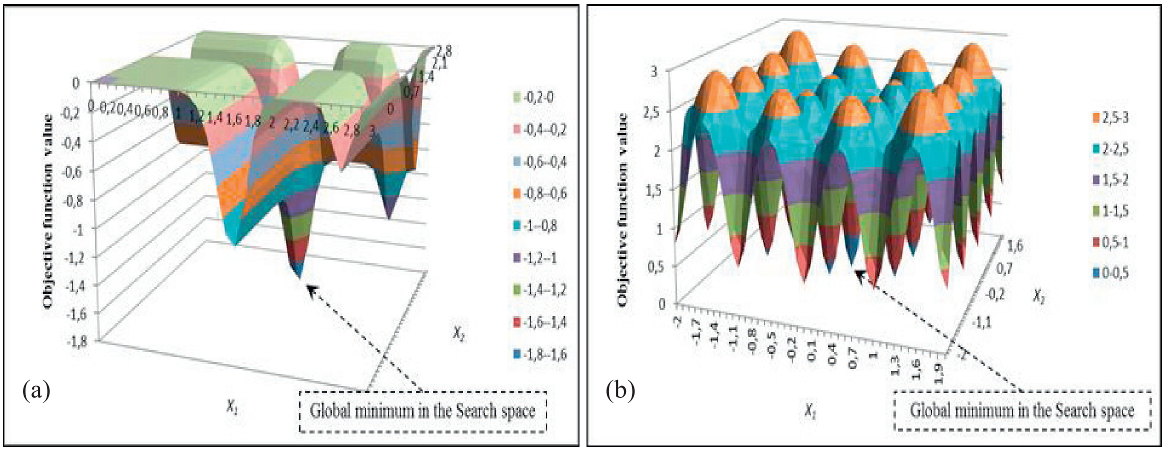

Fig. 3. (a) Objective Function - Michalewicz´s Function; (b) Objective Function - Ackley´s Function.

4. Evaluation of optimization experiments

The behavior of optimization algorithms is random, so we had to perform many optimization experiments to identify the pure nature of the optimization algorithms. Considering the number of simulation experiments we can divide the number of simulation experiments as follows:

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

We specified the same conditions which had to be satisfied for each optimization method, e.g. the same termination criteria, the same search space where the optimization method can search for the global optimum. If the optimization method has the same parameters as another optimization method, we set up both parameters with the same boundaries (same step, low and high boundaries).

The second module is focused on testing the behavior of optimization method in terms of setting the parameters for the optimization method. The user can set up the parameters of a selected optimization method, low and high boundaries of the optimization method parameters, number of replications, and export the objective function chart to image. The results of optimization series are exported to MS Excel workbook. Excel was selected because of its wide usage, specifying formulas, visualizing the data to charts, etc. After finishing the series boxplot characteristics are calculated (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) and three boxplot charts are generated - Best objective function value, Range of provided function objective values during the simulation experiments, and Number of experiments required to find global (local) optimum. Visualization can help the user to find a suitable setting of optimization method more quickly.

We have to propose evaluation techniques which express the failure of the optimization method in different ways due to the large volume of data (over 4 billion simulation experiments). Each criterion value is between [0, 1]. If the failure is 100[%] the criterion equals 1 therefore we try to minimize all specified criteria. The user can set up the weight of each criterion. Other parameters necessary for evaluation are automatically loaded from simulation optimization results.

4.1. Optimization Method Success

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

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

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

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

where i denotes the index of one series, f_{1_i} denotes the value of the first criterion, s denotes the number of performed series. The average optimization method success of finding the optimum of testing functions is shown in Fig. 4.

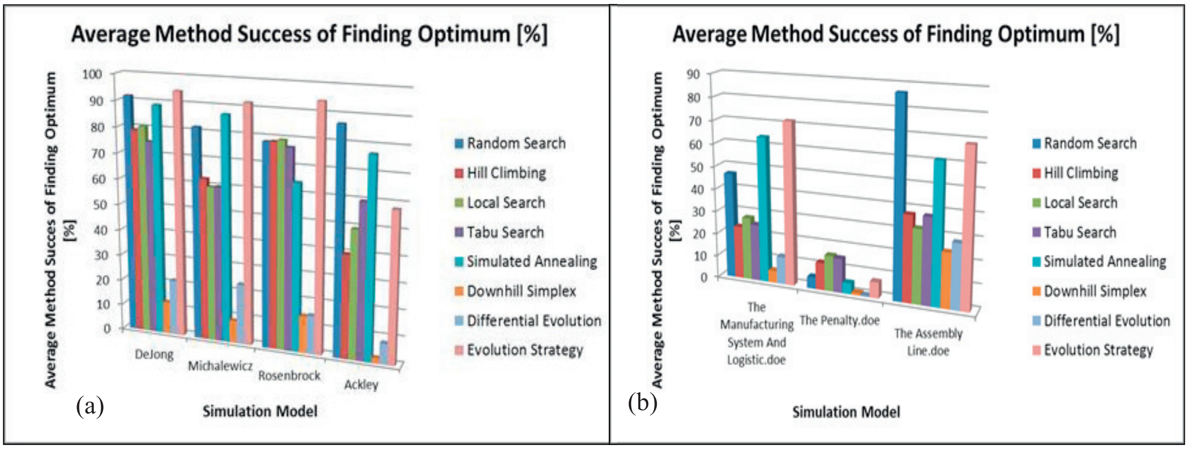

Fig. 4. (a) Average Optimization Method Success – Simulation Optimization Results of Testing Functions; (b) Average Method Success – Simulation Optimization Results of Discrete Event Simulation Models.

The Evolution Strategy and Simulated Annealing are successful optimization methods. Random Search also achieves good results. It was affected by doing many simulation experiments by this method in a small search space (we have to evaluate each possible solution in all the search spaces to obtain the optimum hence the search space cannot be too huge). Random Search was not successful in the case of the Penalty model because of the larger search space. The Penalty discrete event simulation model has a complicated objective function landscape. The optimization methods were not successful in finding the optimum, because the area around the optimum is straight and the method could not obtain information about raising or decreasing the objective function terrain.

These charts also contain bad settings, therefore we separated the bad series from the good series. The next chart contains the filtered series with the best found first criterion value only (the first criterion equals zero, so the optimum was found in each optimization experiment).

The percentage of absolutely successful series compared to all performed series is shown in Fig. 5. The Evolution Strategy has problems with the multimodal Ackley´s function (the method was affected by the number of individuals randomly chosen from the population for the tournament).

This setting affects the exploration (the procedure which allows search operations to find new and maybe better solution structures). The opposite of exploration is the exploitation of the search space (the process of improving and combining the traits of the currently known solutions, as done by the crossover operator in evolutionary algorithms, for instance).

The behavior of Stochastic Hill Climbing, Stochastic Local Search and Stochastic Tabu Search is similar (the similar pseudo gradient principle).

Substandard results were achieved with the Downhill Simplex method. This optimization method works by calculating the points of the centroid (center of gravity of the simplex). We have to modify this optimization method in such a way that it is applicable for discrete event simulation optimization purposes where the step in the search space is defined. We use the rounding of coordinates of the vector (new calculated point) to the nearest feasible coordinates in the search space and this leads to deviation from the original direction. We performed other simulation experiments with smaller steps and the success of finding the optimum was higher than before. This problem can be solved by using a calculation with the original points and the objective function value will be calculated by the approximations of the objective value of the nearest feasible points in the search.

Differential Evolution uses the elitism strategy in our case (the copying of identical individuals suppresses the diversity of new promising individuals – exploration vs. exploitation. Random Search looks successful, but there were only two possible settings – generating the same individual possibility.

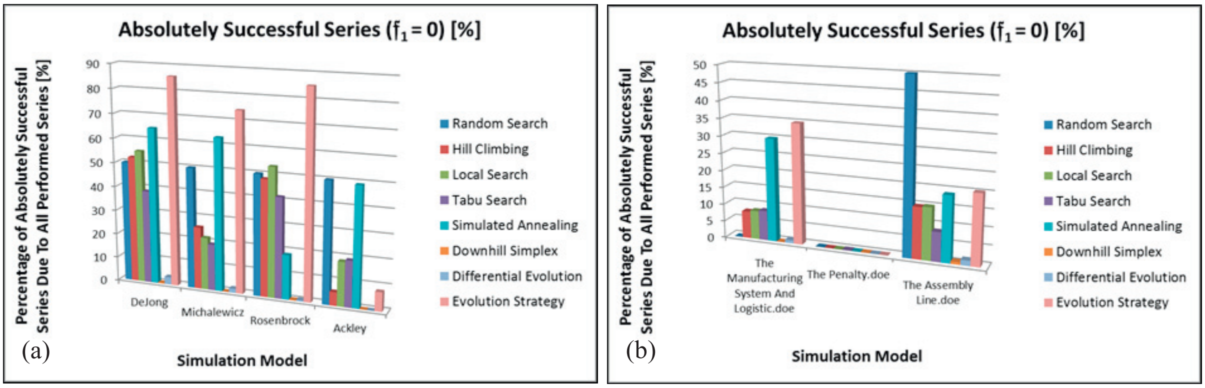

Fig. 5. (a) Percentage of Absolutely Successful Series Compared To All Performed Series - Testing Functions; (b) Percentage of Absolutely Successful Series Considering All Performed Series - Discrete Event Simulation Models.

4.2. The Difference between Optimum and Local Extreme

The second criterion is useful when there is no series which contains any optimum or the solution whose objective function value is within the tolerance of optimum objective function value (the first criterion equals zero in this case). This function evaluates the difference between the objective function value of the best solution found in the series and the optimum objective function value. The effort is to minimize the second criterion. The second criterion is calculated using the formula:

$$f_2 = \frac{\left|F\left(\mathbf{X}^*\right) - F\left(X_{\mathrm{Best}}\right)\right|}{\left|F\left(\mathbf{X}^*\right) - F\left(X_{\mathrm{Worst}}\right)\right|} \tag{9}$$

where $F\left(\mathbf{X}^*\right)$ denotes the objective function value of the global optimum of the search space; $F(X_{\mathrm{Best}})$ denotes the objective function value of the best solution found in a concrete series; $F(X_{\mathrm{Worst}})$ denotes objective function value of the worst solution of the search space.

The difference between the optimum and the local extreme is shown in Fig. 6. The charts contain only series where the first criterion equals zero (no optimum was found in the series). The average of the second criterion is shown for each optimization method – these values express the failure of the optimization method.

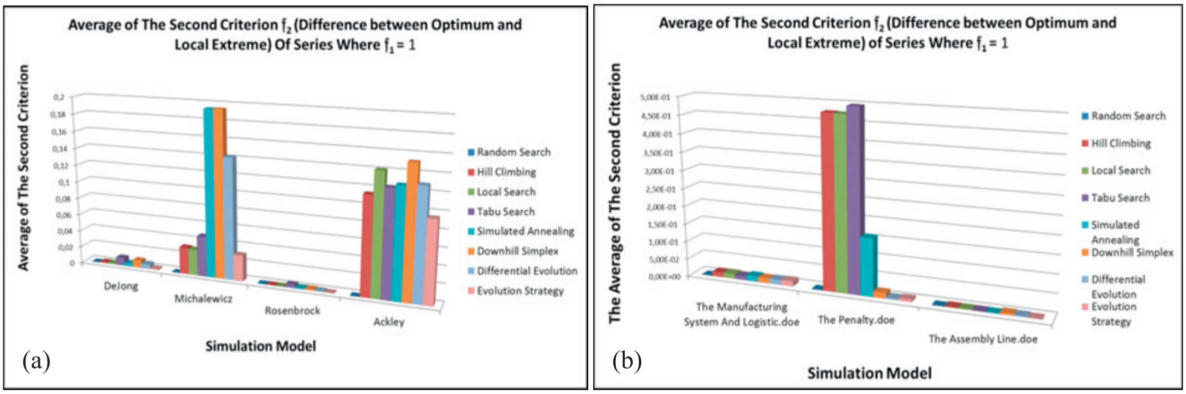

Fig. 6. (a) Average of the Second Criterion (Difference between Optimum and Local Extreme) - Testing Functions; (b) Average of the Second Criterion - Discrete Event Simulation Models.

4.3. The Distances of Quartiles

The third criterion expresses the distance between quartiles of a concrete series. Weights are used for evaluation purposes. These weights penalize the solutions placed in quartiles. Values of the weights were 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:

$$f_3 = \frac{\left|Q_1 - F\left(\mathbf{X}^*\right)\right| + w_{4f_3}\left|Q_1 - Q_2\right| + w_{3f_3}\left|Q_2 - Q_3\right| + w_{2f_3}\left|Q_3 - Q_4\right| + w_{1f_3}\left|Q_4 - Q_5\right|}{\left|F\left(\mathbf{X}^*\right) - F\left(X_{\text{Worst}}\right)\right|} \tag{10}$$

where $F\left(\mathbf{X}^*\right)$ denotes the objective function value of the global optimum of the search space; w_{4f3} denotes the weight (penalty) of objective function values between sample minimum Q_1 and lower quartile Q_2; w_{3f3} denotes the weight of objective function values between lower quartile Q_2 and median Q_3; w_{2f3} denotes the weight of objective function values between median Q_3 and upper quartile Q_4; w_{1f3} denotes the weight of objective function values between upper quartile Q_4 and largest observation - sample maximum Q_5; $F(X_{\text{Worst}})$ denotes objective function value of the worst solution (element) of the search space. The evaluation of optimization experiments using the third criterion is shown in Fig. 7.

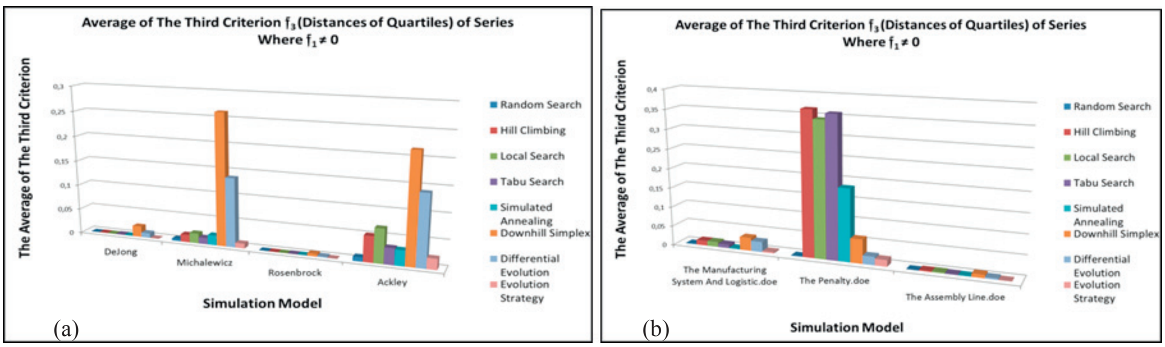

Fig. 7. (a) Average of the Third Criterion (Distances of Quartiles) - Testing Functions; (b) Average of the Third Criterion - Discrete Event Simulation Models.

The effort is to minimize the value of the third criterion. If the first criterion equals zero then the third criterion equals zero. The Downhill Simplex optimization method provided the worst optimization results of all tested optimization methods due to rounding of the coordinates. Pseudo gradient optimization methods found solutions of similar quality. Simulated Annealing provides a worse solution than the Evolution Strategy.

4.4. 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 – see Fig. 8.

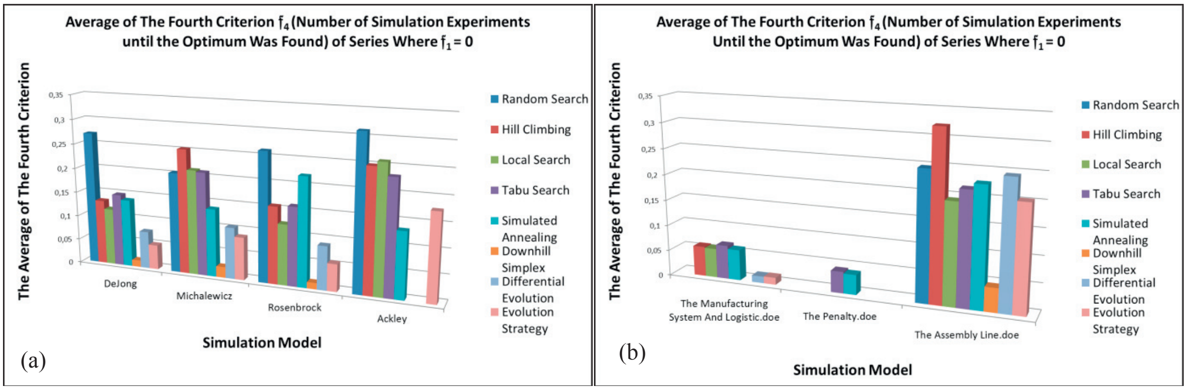

Fig. 8. (a) Average of the Fourth Criterion (Number of Simulation Experiments until the Optimum Was Found) - Testing Functions; (b) Average of The Fourth Criterion - Discrete Event Simulation Models.

The fourth criterion when the objective function is minimized can be formulated as follows:

$$f_4 = \frac{|Q_1 - 1| + w_{4f_4}|Q_1 - Q_2| + w_{3f_4}|Q_2 - Q_3| + w_{2f_4}|Q_3 - Q_4| + w_{1f_4}|Q_4 - Q_5|}{m_{\tilde{X}}} \tag{11}$$

where w_{4f_4} denotes the weight of number of simulation experiments until the optimum was found between sample minimum Q_1 and lower quartile Q_2; w_{3f_4} denotes the weight of number of simulation experiments until the optimum was found between Q_2 and median Q_3; w_{2f_4} denotes the weight of number of simulation experiments until the optimum was found between Q_3 and upper quartile Q_4; w_{1f_4} denotes the weight of number of simulation experiments until the optimum was found between Q_4 and largest observation Q_5; $m_{\tilde{X}}$ denotes the number of feasible solutions in the search space.

5. Conclusion

The goal of the research is to compare selected optimization methods - Random Search, Stochastic Hill Climbing, Stochastic Tabu Search, Stochastic Local Search, Downhill Simplex, Simulated Annealing, Differential Evolution and Evolution Strategy. The success of heuristic optimization methods depends on the objective function landscape. Evolution Strategy is a suitable optimization method for all the tested objective functions (little propensity to bad methods for tuning parameters). This optimization method achieves good values for the second criterion (distance between found local optimum and global optimum or VTR). The alternative to Evolution Strategy is Simulated Annealing. Simulated Annealing has the ability to escape from the local extreme thanks to the implemented approach of setting the temperature to the initial temperature. The strategy of Random Search is simple and effective with a small search space, but if the search space is huge (NP-hard) we can say it is lucky to find the optimum.

Pseudo-gradient optimization methods (Stochastic Hill-Climbing, Stochastic Local Search, Stochastic Tabu Search) achieve almost the same results for the simple objective function landscape.

Differential Evolution uses the elitism strategy (faster finding of a feasible solution but not the finding of the global optimum). The range of provided simulation optimization results using this optimization method is better than the optimization methods based on pseudo-gradient searching.

Acknowledgements

This paper was created with the subsidy of the project SGS-2012-063 “Integrated production system design as a meta product with use of a multidisciplinary approach and virtual reality” carried out with the support of the Internal Grant Agency of University of West Bohemia and with the subsidy of the Motivation System (POSTDOC) of University of West Bohemia. The paper uses the results of the project CZ.1.07/2.3.00/09.0163.

References

[1] V. Votava, Z. Ulrych, M. Edl, V. Trkovsky, M. Korecky, Analysis and Optimization of Complex Small-lot Production in new Manufacturing Facilities Based on Discrete Simulation, in: Proceedings of 20th European Modeling & Simulation Symposium EMSS 2008, Campora San Giovanni, 2008, pp. 198-203.

[2] P. Kopeček, Heuristic Approach to Job Shop Scheduling, in: DAAAM International Scientific Book 2012, Published by DAAAM International, Vienna, Austria, 2012, pp. 573- 584.

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

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

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

[6] J. Tvrdík, „Evolutionary algorithms - Textbooks (Czech language Evoluční algoritmy - učební texty), University of Ostrava, 2004. Available: http://prf.osu.cz/doktorske_studium/dokumenty/Evolutionary_Algorithms.pdf, Accessed: 2008-02-06.

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

[8] X. Zeng, W.-K. Wong a S. Y.-S. Leung, „An operator allocation optimization model for balancing control of the hybrid assembly lines using Pareto utility discrete differential evolution algorithm," Computers & Operations Research, Vol. 39, Issue. 5, May 2012, pp. 1145-1159.

[9] L. Xiangtao a Y. Minghao, „An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," Advances In Engineering Software, Vol. 55, January 2013, pp. 10-31.

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

[11] F. Koblasa, F. Manlig, J. Vavruška, Evolution Algorithm for Job Shop Scheduling Problem Constrained bythe Optimization Timespan, Applied Mechanics and Materials, 2013, pp. 350-357.

[12] H. Pohlheim, „GEATbx: Example Functions, Available: http://www.geatbx.com/docu/fcnindex-01.html#P204_10395. Accessed 2011-11-20.