Simulation-based Optimization; Methods and Practical Application

Ewa Niewiadomska-Szynkiewicz[1,2]

[1] Warsaw University of Technology, Institute of Control and Computation Engineering, Warsaw, Poland, email: ens@ia.pw.edu.pl
[2] Research Academic Computer Network (NASK), Warsaw, Poland, e-mail: ewan@nask.pl

Abstract. The paper is concerned with computational research for complex systems. The simulation-based optimization approach, which is widely used in applied science and engineering, is formulated and discussed. The numerical techniques that optimize performance of system by using simulation to evaluate the objective value are reviewed. The focus is on random search and metaheuristics. The practical example - application of simulation optimization to calculate the optimal decisions for controlling the river-basin reservoir system during flood period is presented and discussed.

1 Introduction

Traditionally, the complex optimization problems are solved using linear and nonlinear techniques, which normally assume that the performance function and the set of admissible solutions are known in analytical form. In many practical contexts, the optimization problems cannot be described analytically due to the natural complexity and uncertainty of the real-life systems. The load of mathematical and practical knowledge put to the model results in numerous formulas whose solution can be obtained only numerically. Computer simulation is a standard tool for understanding and predicting the behavior of a system under the influence of various realistic and stochastic input scenarios. Although simulation has traditionally been viewed as an approach of last resort, recent advances in computer hardware and software have made it one of the most popular technique to attack many real-life problems. In recent years we can observe a rapid growth of simulation-based optimization (or simulation optimization) which merges optimization and simulation technologies, [3, 5, 18]. We can formulate the definition:

Definition. *Simulation optimization is a search or optimization technique that uses the simulation experiment to evaluate the expected performance of the system for each set of decision variables (system inputs).*

Let us consider an example:
Example: *A river-basin system. It consists of a net of rivers and retention reservoirs located on tributaries to the main river. Our goal is to develop the system controlling the operation of the reservoirs during flood periods. The idea is to achieve the coordination*

of reservoirs in minimizing the flood damage created by a flood wave passing through the river basin. We can formulate the problem to be optimized - flood damage minimization:

$$\min_{x \in X} J(x) \tag{1}$$

where x denotes vector of outflows from the reservoirs (controls) and J flood damage related to the high water levels in the considered river-basin. To calculate the optimal values of outflows from the reservoirs the expected damages related to the proposed decisions have to be computed. Taking into account the description of the system to be controlled (nonlinear, complex models of flow transformation through the river basin, limited knowledge regarding future inflows, constraints on the reservoirs capacity and releases, many decision units and, at the same time, many different individual objectives, etc.) we cannot define the analytical form of the performance function. The only viable option is to perform the simulation experiment for forecasts of the future inflows to the system. So for every optimization step it is necessary to calculate the expected value of the performance, i.e., J in (1). So, the simulation optimization scheme as presented in Fig. 1 is proposed to solve the problem.

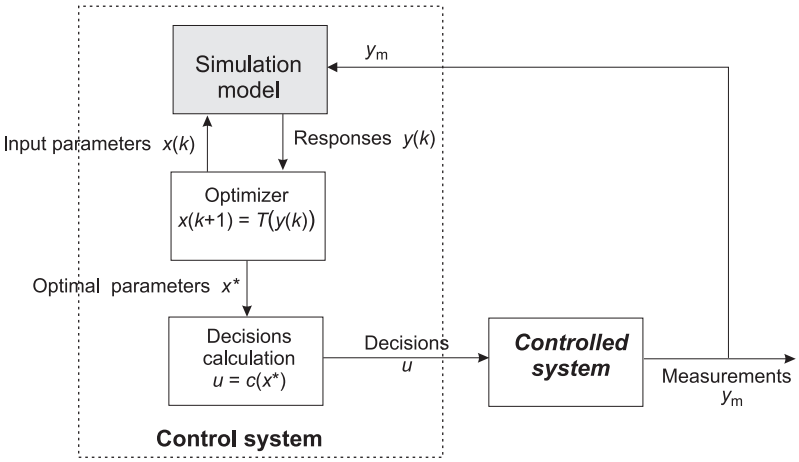

Figure 1. Simulation optimization scheme for control decisions calculations in operational control.

2 Simulation optimization problem formulation and discussion

In the context of simulation optimization, a simulation model is a function, $S : \mathbb{R}^n \to \mathbb{R}^m$, mapping input parameters, $x \in \mathbb{R}^n$, to outputs, $y \in \mathbb{R}^m$. The explicit form of the function S is usually unknown. Its numerical representations (sets of y) that are the results of simulation experiments performed for various input parameters x are called *response surfaces*. We can formulate the general problem of optimization through embedded simulation

$$\min_x \left[f(x) = \Psi(x, S(x)) \right], \qquad x \in D_x \tag{2}$$

In the above formula $\Psi : \mathbb{R}^{n+m} \to \mathbb{R}$ is a performance measure calculated for simulator responses $y = S(x)$ for input variables x, and D_x specifies the constraints on the input variables. So, in this context the role of the simulator (simulation model) is to transform input parameters x into outputs y used to calculate the performance measures.

The optimization problem (2) is usually nonlinear, often non-convex and gradient information is rarely available. The features that complicate the task are the demand that the solution is globally optimal and the constraints may be formulated both with decision variables (simulator inputs x) and the responses y. If the constraints depend only on input variables x the feasible trial solution can be checked before the simulation execution, otherwise the feasibility of the new trials is not known before running the simulation experiment. Additionally the constraints provided explicitly are accompanied by a number of implicit constraints – defined on the internal variables of the modelled system. Hence, it is difficult or even impossible to asses whether the domain D_x is convex and compact.

As abstractions of real-life systems operation, simulators have limits of credibility. Such limits may result from simplifications in the mathematical modelling and its computer implementation, the omission of some possible values of inputs and the random effects in the considered system (for example uncertain system – environment interactions). Assume that vector $V = [\vartheta_1, \ldots, \vartheta_m]$ represents the amalgamation of many individual random effects in the simulator. The simulator response for decision variables x and random effects V are calculated as $y = \tilde{S}(x, V)$, where $\tilde{S}$ denotes a simulation model considering the random effects in the simulated system. The sample realization of the performance function calculated from running the simulation is denoted by q. Let us formulate the stochastic simulation-based optimization problem

$$\min_x \left[f(x) = E\left\{ q(x, V) = \Psi(x, \tilde{S}(x, V)) \right\} \right], \qquad x \in D_x \tag{3}$$

In the above formula f represents an average performance measure over all possible values of V at the specified x.

We can distinguish two situations concerned with the type of influence on the system by the inputs x:

A: Inputs x effect directly the outputs y and does not influence the random vector V. The performance function in (3) can be calculated as follows:

$$f(x) = E\{q(x, V)\} = \int_{D_v} q(x, v) p(v) \mathrm{d}v \tag{4}$$

where $p(v)$ denotes the density of probability distribution of V, and D_v the domain of V.

B: Inputs x enter via their effect on the probability distribution of V. In this case the performance function in (3) can be calculated as follows:

$$f(x) = E\{q(x, V)\} = \int_{D_v} q(x, v) p(v|x) \mathrm{d}v \tag{5}$$

where $p(v|x)$ denotes the density of probability distribution of V.

It is useful to distinguish both presented types of input parameters when gradient optimization methods are considered. It is proved in [18] that in the case of variant A the infinitesimal perturbation analysis (IPA) method is used to estimate the stochastic gradient $\nabla \hat{f}(x)$[1], in the case of B the likelihood ratio method (LR) is applied, for both types of inputs (variant A and B) IPA/LF is used.

IPA: $\frac{\partial q(x,V)}{\partial x}$,

LR: $q(V)\frac{\partial \log p(V|x)}{\partial x} = \frac{q(V)}{p(V|x)}\frac{\partial p(V|x)}{\partial x}$, $p(V|x) > 0$,

LR/IPA: $q(x,V)\frac{\partial \log p(V|x)}{\partial x} + \frac{\partial q(x,V)}{\partial x}$.

In conclusion, simulation optimization is generally complex but seems to be the only viable option to solve many practical engineering problems. The restrictions are caused by the demands on computation time. The efficient optimization process requires a large number of iterations and the system performance evaluations. For every iteration of the optimization algorithm we have to perform time-consuming simulation experiment (see Fig. 1), which for some sets of inputs does not always finish successfully.

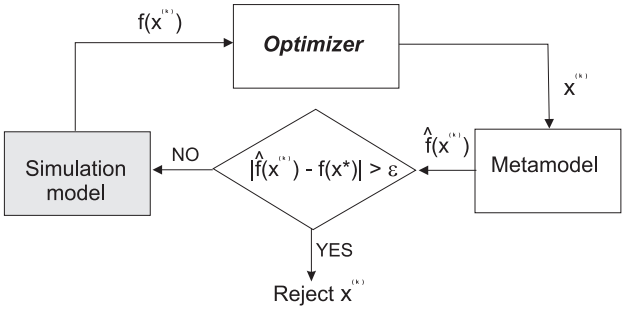

Figure 2. Metamodel application to optimization; x^* – the best (current) solution, Ψ – the performance obtained based on simulation, $\hat{\Psi}$ – the performance obtained by evaluating the metamodel, ϵ - assumed accuracy.

The direction, which should bring benefits is parallel computing where the whole task is partitioned between several processors. Parallel implementation allows us to reduce the computation time, and to execute large simulation programs which cannot be put on a single processor.

Another approach to reduce the computational load and at the same time the time of calculations, is the use of metamodels. A metamodel is used in the optimization to estimate performance in (2) or (3) instead of the simulation model. It approximates the response surface of S function and the performance measure Ψ corresponding to the simulation model. It is often used as a filter the goals of which are to predict the performances for new trial solutions, compare them with the current best known one and eliminate low quality solutions from further consideration, see Fig. 2. In such approaches the decision concerned with trial points rejection is a trade off between the speed and accuracy of the search. The popular techniques used to build metamodels are: linear and non-linear regression, neural networks, etc. Unfortunately, the metamodel

[1]This expression is called a stochastic gradient because it is noisy measurement of the gradient (it depends on the random effects V).

implementation usually requires a lot of measurements and has proved to be costly in many real-life applications.

3 Practical Approaches to Simulation Optimization

Many optimization techniques that could be employed for solving complex optimization problem (3) have been reported. The classical approaches for optimizing simulations and their applications are discussed in [3, 5, 18].
We can distinguish several approaches for optimizing simulations:

- stochastic approximation (gradient based approaches),
- sample path optimization,
- response surface methodology,
- deterministic search methods,
- random search methods,
- heuristics and metaheuristics.

A *Stochastic approximation* is the well known gradient search method that is similar to steepest descent gradient algorithm. The procedure requires a gradient estimation. The computer simulation is applied to obtain estimates of the gradient The simplest way is to use finite differences and multiple simulations to calculate the derivatives of the expected performance of the system. Another approach is to apply perturbation analysis (IPA) or likelihood ratio method (LR). These techniques require only single simulation run at one set of inputs but involve differentiating in equations (4) or (5) directly. Is seems that the simulation outcome has to contain gradients evaluations. Both approaches mentioned have advantages and disadvantages that are discussed in detail in [5] and [18].

For every iteration stochastic gradient algorithms need a simulation run to calculate the gradient value (K iterations require at least K experiments). In the *sample path* method the original problem is converted into an approximated deterministic problem. The approximation $\hat{f}$ of objective f is calculated based on simulations performed for random generated set of independent observations V, i.e., $V_1, V_2, \ldots V_M$, $M \neq K$ known as *sample path*. Then the standard optimization algorithms are applied to locate the optimal solution. In the case when cumulative distribution function of the random vector V does not depend on inputs x optimization problem (3) is transformed to:

$$\min_x \left[\hat{f}(x) = \frac{1}{M} \sum_{i=1}^{M} q(x, V_i) \right] \tag{6}$$

A response surface methodology is a sequential strategy based on local approximation $F(x, \alpha)$ of the performance f in the neighborhood $D_x^{(k)}$ of x, where the parameters α are calculated using simulations, k is a iteration number. The minimal value of $F(x, \alpha^{(k)})$ is calculated. The process is repeated unless the acceptable solution is found. The idea is similar to approaches with metamodels application but in this method the metamodels are used to characterize the objective function in the local, currently explored area.

Standard deterministic search techniques [16], i.e. algorithms developed by Hook and Jeeves, Rosenbrock or nonlinear simplex (as Nelder and Mead) or complex methods can be applied to solve non-differentiable simulation optimization problems.

As result of the growing possibilities of modern computers, we can observe increasing interest in the development of the global algorithms that are concerned with the

computation and characterization of global optima of non-convex functions. During the last decades many theoretical and computational contributions helped to solve multiextreme problems. Global optimization methods are widely used in many industrial and scientific applications. Their approach to simulation optimization is based on viewing the simulation model as a black box function evaluator. These approaches are flexible, robust and less demanding properties of the problem. Many of them are constituted by the stochastic algorithms typically based on random search, [18, 19]. Monte Carlo techniques, multi-start local search and Adaptive search methods, like Controlled Random Search CRS [1, 2, 17] belong to this category. Most of the global techniques utilize heuristics and do not guarantee the optimum solution, but rather provide a reasonable solution in a reasonable time. Genetic Algorithms GA [8, 9, 11], Evolutionary Strategies ES [4, 11, 12], Simulated Annealing SA [6, 10, 12], clustering techniques [19] and Tabu search [7] are all of a heuristic nature. Nowadays application of heuristics and metaheuristics to simulation optimization problems is very popular [3].

4 Case study: Vistula river reservoir system

The presented case study is related to hierarchical control structure for flood operation in the Upper Vistula river-basin system in the Southern part of Poland. The system consists of three reservoirs *Tresna*, *Dobczyce* and *Rożnów*, located on tributaries to the Vistula river, and of three uncontrolled side inflows. Simulation-based optimization is applied by the central authority of the system to calculate parameters coordinating the management of the reservoirs.

4.1 Problem formulation

The optimization problem to be solved was described in the example presented in section 1. It is defined as the problem of minimization the flood damage J related to the peak flows at the measurement points in the whole river system:

$$J(\boldsymbol{Q}_{[t_0,t_f]}) = \sum_{k=1}^{K} \beta_k \max(Q_k^{cul} - Q_k^{limit}, 0) \tag{7}$$

In the formula (7) Q_k^{limit} denotes highest safe discharge at the k-th damage center (with respect to protection of banks and flood damage), Q_k^{cul} the peak discharge, $Q_k^{cul} = \max_{t\in[t_0,t_f]} Q_k(t)$ for $k = 1, \ldots, K$, where $[t_0, t_f]$ is the control horizon and $Q_k(t)$ flow at the k-th damage center at time t; β_k denotes the weighting factor related to the flow at the k-th damage center (different points have different importance); and K the number of damage centres.

The hierarchical control mechanism for reservoirs management was investigated. This mechanism is based on the use of the repetitive optimization of the outflow trajectories, using the predicted inflows – forecasts; see [14, 15] for details. It incorporates two decision levels each as presented in Fig. 3: the upper level with the control center (coordinator) and the local level formed by the operators of the reservoirs. Within this structure, the central dispatcher performs an analysis of possible future scenarios of the flood and determines the optimal vector of coordinating parameters a influencing local operator decisions about

the outflows from the reservoirs solving the optimization problem $\min_{\boldsymbol{a}\in A} J(\boldsymbol{a} \in A)$, with J defined in (7). The local decision rules are designed in such a way that a central authority, the coordinator, may adjust them in the process of periodic coordination, so as to achieve the coordination of reservoirs in minimizing global damage. It is assumed

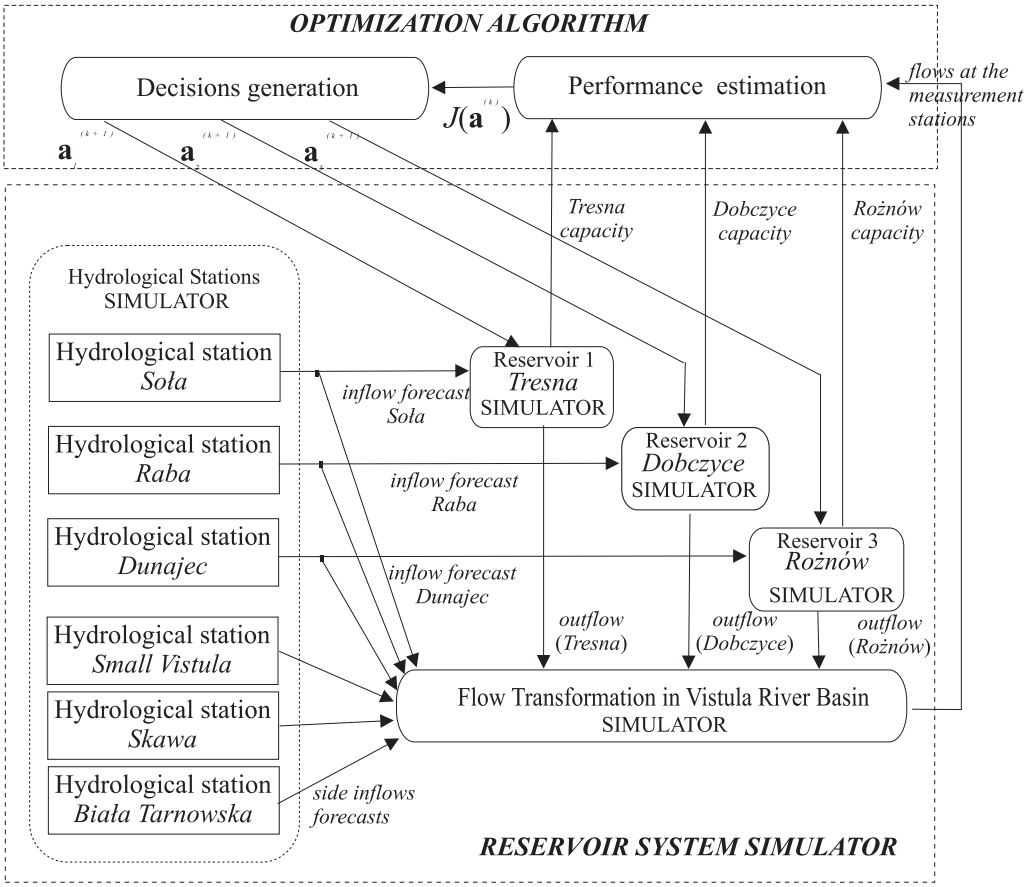

Figure 3. Two-level control structure for controlling the Vistula river-basin system.

that the vector $\boldsymbol{a}_i$ of coordinating parameters for the i-th reservoir, $i = 1, 2, 3$, considered at time t_l is related to the weighting function $\alpha_i(t)$ defined as follows: $\alpha_i(t) = 1 + (c_i - 1) \cdot \mathbf{1}(t - T_i^*)$, i.e. $\alpha_i(t) = 1$ for $t \in [t_l, T_i^*)$ and $\alpha_i(t) = c_i$ for $t \in [T_i^*, t_f]$. Finally, the vector $\boldsymbol{a}_i$ is given as $\boldsymbol{a}_i = [c_i, T_i^*]$.

Calculated parameters are used to modify the local performance measures $q_i = \max(x_i(t))$ for $t \in [t_l, t_f]$, i.e., the damage created by high water levels directly downstream of the reservoirs; where, $t \in [t_l, t_f]$ denotes local level optimization horizon. Hence, the decision problem of the i-th local reservoir operator at time t_l is as follows:

$$\min_{u_i} \left[q_{i,mod}(u_i(\cdot), \boldsymbol{a}_i) = \max_{t\in[t_l,t_f]} (u_i(t) \cdot \alpha_i(t)) \right] \tag{8}$$

In the calculation process performed by the central operator two phases can be distinguish: optimization and simulation. At every optimization step the value of damages is estimated based on the outputs of the simulation process. The optimization is realized as follows: after assuming certain values of parameters $\boldsymbol{a}$, simulation of the reservoirs operation and flow transformation in the whole river basin until the predicted end of the flood t_f is performed. The implicit constraints on the capacity w and outflows x from the reservoirs have to be concerned during simulation process: $w_{i_{min}} \leq w_i(t) \leq w_{i_{max}}$, where $w_{i_{min}}$ and $w_{i_{max}}$ minimal and maximal capacity, i–th reservoir number, $x_{i_{min}}(w_i(t)) \leq wx_i(t) \leq x_{i_{max}}(w_i(t))$, $x_{i_{min}}$ and $x_{i_{max}}$ minimal and maximal outflow, and additional for the desirable final storage of the reservoirs: $w_i(t_f) = w_f$. The simulation responses are expected flow trajectories in the considered measurement stations in the river basin. Then, the value of the overall performance index J related to the given vector a is computed. The optimization simulation algorithm for optimal coordinating parameters calculation is presented in Fig. 4.

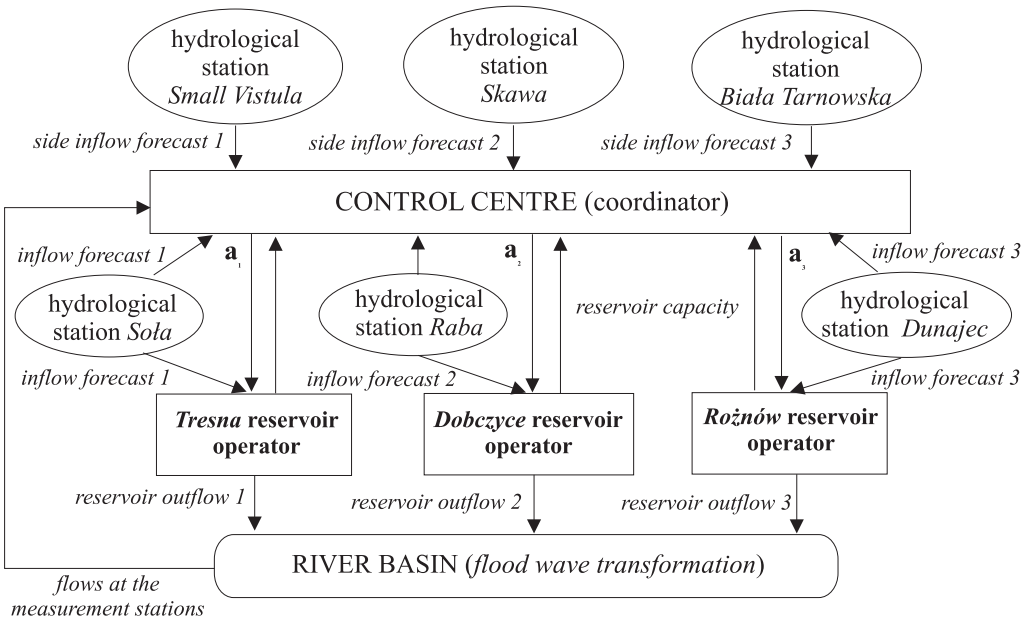

Figure 4. Simulation-based optimization algorithm for optimal coordinating parameters calculations (Vistula river-basin system).

Numerical experiments were performed for a set of data containing historical floods and several hypothetical, so-called scenarios. The central operator performed optimization of six parameters forming vector a, using four heuristic optimization methods - deterministic and stochastic techniques. The first heuristic applied was deterministic, direct search algorithm - Downhill Simplex Algorithm NM developed by Nelder and Mead. Next, global optimization techniques – random search and metaheuristics were tested. Many calculations were performed for every method. The question was how the global algorithms influence the optimization results and thus influence the issues of the operation of a multireservoir system during flood. The most interesting numerical results

obtained for two historical floods that occurred in 1970, 1972 and one hypothetical scenario of the flood are collected in Table 1. The table presents the best (i.e., the lowest) and the worst (i.e., the highest) optimal values of J obtained during 10 runs of each optimization algorithm and the reduction of the performance index with respect to the Downhill Simplex Algorithm NM (Nelder-Mead's).

The available numerical results indicate that the considered global optimization algorithms, i.e., ES, CRS2 and CRS3 enable improvement in relation to the standard NM simplex algorithm. In most cases, the best results were obtained by ES but the time required to compute a solution was longer than in CRS methods. The CRS3 method provided better results compared to CRS2. However, the reduction of cost with respect to CRS2 method was not very big. In general, the global algorithms can improve the efficiency of the control system being considered.

Table 1. Flood control optimization - different methods (A – the best result, B – the worst result; /* – reduction of criterion with respect to NM method.

opt.	SCEN		1970		1972	
method.	A /*	B	A /*	B	A /*	B
NM	1587	1587	1989	1989	773	773
CRS2	1574 / 0,81%	1587	1892 / 4,87%	1907	765 / 1,03%	768
CRS3	1574 / 0,56%	1583	1877 / 5,61%	1892	764 / 1,17%	770
ES	1515 / 4,51%	1572	1773 /10,86%	1792	722 /10,81%	770

5 Summary

In this paper a brief review was made of the algorithms for solving simulation-based optimization problems. The focus was on global techniques (random search and metaheuristics). The results of the application of the random search and metaheuristic approaches to solve complex real-world optimization problems were presented and discussed. The algorithms for simulation optimization usually require a large number of iterations and the performance evaluations. Each iteration is costly in the case of optimization through embedded simulation Because of the complexity of the considered simulation model, it is quite often inherently difficult to find a good estimate of the solution with a reasonable number of function evaluations and calculation time. As was mentioned in section 2 the direction, which should bring benefits is parallel and distributed computing where the whole task is partitioned between several processors or machines, as described in [13].

Bibliography

[1] M.M. Ali and C. Storey. Modified controlled random search algorithms. *International Journal of Computer Mathematics*, 54:229–235, 1995.

[2] M.M. Ali, A. Törn, and S. Vittanen. A numerical comparison of some modified controlled random search algorithms. *Journal of Global Optimization*, 11:377–385, 1997.

[3] J. April, F. Glover, J.P. Kelley, and M. Laguna. Practical introduction to simulation optimization. In *Proceedings of the 2003 Winter Simulation Conference*, pages 71–78, 2003.

[4] T. Baeck, D.B. Fogel, and Z. Michalewicz. *Evolutionary computation 2: Advanced algorithms and operators*. Institute of Physics Publishing, Bristol, UK, 2000.

[5] J. Banks, editor. *Handbook of simulation*. John Wiley & Sons, Inc, New York, 1998.

[6] A. Dekkers and E. Aarts. Global optimization and simulated annealing. *Mathematical Programming*, 50:367–393, 1999.

[7] F. Glove. Tabu search – part 1 and part 2. *ORSA Journal on Computing*, 1, 2:190–206, 4–31, 1989, 1990.

[8] D.E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Pub. Co., Massachusetts, 1989.

[9] D.E. Goldberg. *The design of innovation*. Kluwer Academic Publishers, Dordrecht, 2002.

[10] R. Horst and P.M. Pardalos. *Handbook of global optimization*. Kluwer Academic Publishers, Dordrecht, 1995.

[11] Z. Michalewicz. *Genetic algorithms + data structures = evolution programs*. Springer-Verlag, Berlin Heidelberg, 1996.

[12] Z. Michalewicz and D.B. Fogel. *How to solve it: Modern heuristics*. Springer-Verlag, New York, 2000.

[13] E. Niewiadomska-Szynkiewicz. Pareallel global optimization for optimal flood control. *Acta Geophysica Polonica*, 47:93–109, 1999.

[14] E. Niewiadomska-Szynkiewicz. Computer simulation of flood operation in multireservoir systems. *Simulation*, 80:101–116, 2004.

[15] E. Niewiadomska-Szynkiewicz, K. Malinowski, and A. Karbowski. Predictive methods for real time control of flood operation of a multireservoir system - methodology and comparative study. *Water Resources Research*, 32:2885–2895, 1996.

[16] W.H. Press, S.A. Tukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C. The art of scientific computing*. Cambridge University Press, Cambridge, 1992.

[17] W.L. Price. Global optimization algorithms for a cad workstations. *Journal of Optimization Theory and Applications*, 55:133–146, 1987.

[18] J.C. Spall. *Introduction to stochastic search and optimization*. John Wiley & Sons, New Jersey, 2003.

[19] A. Törn and A. Žilinskas. *Global optimization*. Springer-Verlag, Berlin, 1989.