Czech University of Life Sciences Prague
Faculty of Environmental Sciences
Department of Water Resources and Environmental Modeling

Doctoral thesis

Application of optimization methods in hydrological modelling

Ing. Michala Jakubcová

Supervisor: doc. Ing. Petr Máca, Ph.D.

Prague 2015

Author's declaration

I hereby declare that this submitted thesis “Application of optimization methods in hydrological modelling” is my own work, all co-authors of the manuscripts are properly named, and only sources listed in the Bibliography were used.

Prague, 10^{th} November 2015

....

Michala Jakubcová

Dedication and acknowledgements

I would like to thank to Petr Máca, who was the best supervisor during my study, and without his help I would not make it that far. He motivated me to sink into the particle swarm optimization, to learn the C++ programming language, to use LaTeX, and many more. He guided me very patiently, supported me, helped me write research papers, and gave me many valuable advices.

Special thanks belong to many of my colleagues from the Department of Water Resources and Environmental Modeling. They all together created very friendly working environment, where science was created next to fun and joy. I namely thank to Pavel Pech for his helpfulness, trust and optimism with which he listens to students and leads the department, and to Helenka Michálková for all administrative matters, and friendly and supportive approach.

I am very grateful to my family for their support throughout my studies. The biggest thanks belong to my beloved husband Pavel Jakubec, who was always very patient, tolerant and helpful, and who encouraged me in my studies.

My research was financially supported by the Internal Grant Agency of the Faculty of Environmental Sciences, Czech University of Life Sciences Prague (Projects no. 20124259, 20134271, 20144227 and 20154219), and by operational costs of the Department of Water Resources and Environmental Modeling.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

Finding the optimal state of reality is one of the basic principles of the world. Atoms are forming in shapes with minimal energy between electrons, molecules are creating optimal crystal structures in terms of energy [114]. One of the biological principle is the *survival of the fittest*, which together with the evolution leads to better adaptation of organisms to their environment [129]. The well-adapted species dominate other organisms in the vicinity, and they become the *optimum of the locality* [143].

Optimization is a process in which the best variant from many possibilities is chosen. It is very important for increasing effectiveness, or decreasing demands of the computational system.

The optimization process could be very complicated. The main challenges, which can make the process of finding the optimal value of a given objective function more difficult, are [144]:

- premature convergence to a local optimum,
- noisy function with no useful information about the gradient of the function,
- unexpected shape of the function with sudden change of the course,
- function with long slight declining or increasing section, which resembles a constant function.

Therefore, it is necessary to wisely choose a suitable optimization method, and devote some time to its modification according to the given problem.

This doctoral thesis is focused on optimization used in hydrological modelling. The optimization methods are widely applied for calibration of model parameters, for predictions of water quality, surface and groundwater runoff, or for meteorological forecasts [36, 116, 137].

The applied and analysed technique chosen within this thesis is method called particle swarm optimization (PSO). It is inspired by behaviour of social organisms in the nature. The main advantages are low number of parameters, which need to be adjusted, and no requirement of knowledge about gradient of the optimized function [73].

Particle swarm optimization algorithm was used in terms of hydrology for detection of relationship between precipitation and runoff from the catchment [23], for estimation of model parameters [50, 70], during calibration of groundwater model [48], or for training the artificial neural networks (ANN) [20, 21, 74].

1.1 Main goals

Particle swarm optimization was analysed within this doctoral thesis due to its advantages. The PSO was successfully used in many real life case studies, and its applicability and efficiency were proved. It is relatively recent optimization technique, and thus, new modifications can be made to improve the optimization ability.

Main goals of the doctoral thesis are following:

- provide a literature review about the particle swarm optimization method with emphasis to its utilization in hydrological modelling,
- create algorithms of different modified versions of PSO with the implementation in C++ programming language,
- propose new algorithm of PSO, and implement it in C++ programming language,
- test the existing PSO modifications with the new proposed variant on chosen benchmark objective functions,
- applied the best PSO algorithms on case studies regarding rainfall-runoff simulations and training artificial neural networks.

This doctoral thesis will extend the range of global optimization techniques. The results will contribute to utilization of PSO method in real-life optimization problems. New algorithms will have high application potential not only in the field of hydrological modelling. Completed algorithms become basis for other research projects, and they will be available for later use.

1.2 Outline of the thesis

Chapter 2 gives an introduction to the optimization process, and provides a comprehensive review about particle swarm optimization method. It describes fundamental equations of PSO, and explains different types of topology. Modifications of the original algorithm are also listed. Next part of this chapter summarizes chosen objective functions, i.e. benchmark problems for testing optimization algorithms, and frequently used objective functions in hydrological modelling. The optimization process is very important in hydrological modelling for estimating the best set of parameters, for meteorological forecasts, or runoff predictions. Therefore, the last part of the chapter mentions several recent hydrological case studies with utilizing the PSO.

Chapter 3 is based on a published research paper, which focuses on comparison of chosen PSO modifications on single objective benchmark problems. In total, 27 PSO variants were tested on 11 uni-modal and multi-modal benchmark functions. The chapter summarizes the main information about PSO technique, lists selected modifications and benchmarks functions, analyses results, and concludes the main findings.

Chapter 4 is based on a published research paper, which focuses on parameter estimation in rainfall-runoff model Bilan using different versions of PSO. A new PSO modification was proposed, which was first tested on 11 benchmark problems. Then, the new proposed variant was applied together with other 3 PSO variants in Bilan model for streamflow simulations on 30 US catchments.

Chapter 5 is based on a submitted research paper, which focuses on combination of artificial neural networks with particle swarm optimization. Integrated hybrid ANN models were developed, where PSO algorithm was used for training the model weights. In total, 150 different ANN models were applied for simulating the standardized precipitation evapotranspiration drought index (SPEI) on 8 US catchments. The main factors influencing the results are discussed in this paper, and recommendations for later use are given.

Chapter 6 summarizes the most important findings presented in detail in Chapters 2-5, and the overall results of the thesis are put into a broader perspective. Chapter 7 contains summary of this doctoral thesis in Czech language.

CHAPTER 2

PARTICLE SWARM OPTIMIZATION IN HYDROLOGICAL MODELLING

2.1 Introduction to optimization

Optimization is a process which serves to find the optimal values of mathematical function. In many cases, the problem is searching for extremes of the function. Thus, the optimal value is the minimum or maximum. The procedure of finding the minimal and maximal value is equivalent. Therefore, the terms *optimization*, *minimization* and *maximization* have the same meaning (i.e. searching for the optimal solution) [10].

The optimization problem is defined by function f, and by n-dimensional search space $\mathbb{R}^n$. The function f is called an *objective*, *error*, or *fitness function*. The problem can be defined as

$$f : \mathbb{R}^n \to \mathbb{R}. \tag{2.1}$$

If the optimization problem is a minimization of the objective function, the algorithm searches for the minimum $\mathbf{X_{min}} \in \mathbb{R}^n$, for which [143]

$$\forall \mathbf{X} \in \mathbb{R}^n : f(\mathbf{X_{min}}) \leq f(\mathbf{X}). \tag{2.2}$$

The main aim of optimization is to find the best set of parameters of the objective function in an acceptable amount of time. This process is very important in many professions, for example for designing chemical plants to obtain maximum production, to approximate data and minimize differences between measured and calculated values, for allocating resources in an industrial and social environment, for planning the time schedule, etc. [10, 93].

The solution of optimization problem can be found through many optimization methods. The distribution of optimization methods by various authors is summarized in the next subsection. Later in the text, the focus is mainly on the particle swarm optimization technique, which is the procedure used in practical applications within this doctoral thesis.

2.1.1 Optimization methods

Optimization techniques can be split into gradient-based methods, single state methods, and population-based methods according to Luke [85]. To the gradient-based methods belong gradient ascent, or Newton's method. The single state methods are for example hill climbing technique, simulated annealing, or tabu search. The population-based methods contain evolution strategies, genetic algorithm, or differential evolution.

Weise [143] divides optimization methods based on speed of optimization to online, and offline techniques. The online method requires high speed, where the computational time for finding the solution is from milliseconds to several minutes. In offline optimization, the computational time is not essential for the user, and thus, it can take several days to find the optimal value.

Wolpert and Macready [146] proposed an optimization theory called no free lunch theorem. They defined it on a finite search space. The main idea of the no free lunch theorem is that all algorithms are equivalent when their performances are averaged across all problems. Therefore, no one can create an algorithm, which will be better than any other existing algorithm.

The optimization methods can be divided into exact methods, heuristics, and meta-heuristics [93]. The exact methods solve optimization problems by searching the entire solution space completely. They include linear programming, dynamic programming, or divide and conquer technique.

The heuristics are used when the way to find the optimal solution is not known. The solution is approximate, often based on estimation, or experience. Fixed point method, Nelder-Mead algorithm, or gradient methods belong to this technique [93].

The meta-heuristics is applied for solving of general problems. It combines objective functions, or heuristics without a deeper insight into their structures. They can be divided into deterministic, and probabilistic methods [93].

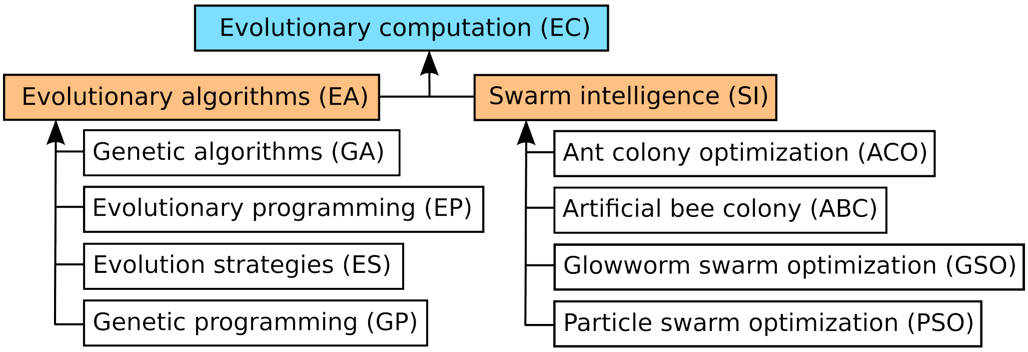
Evolutionary computation (EC)
Evolutionary algorithms (EA)
Swarm intelligence (SI)
Genetic algorithms (GA)
Evolutionary programming (EP)
Evolution strategies (ES)
Genetic programming (GP)
Ant colony optimization (ACO)
Artificial bee colony (ABC)
Glowworm swarm optimization (GSO)
Particle swarm optimization (PSO)

Figure 2.1: Simplified system of evolutionary computation technique (adapted from [143])

Deterministic algorithms are used in cases, where exists a clear relationship between characteristics of possible solutions and their applicability to a given problem. Probabilistic algorithms are used when the relationships are not defined, they are complicated, or the dimension of the search space is too high [93].

Probabilistic methods are based on populations. In this approach, many individuals, which represent possible solutions of the objective function, are stored in the memory. Evolutionary computation (EC) is one of the largely explored probabilistic method, and the simplified system of this technique is depicted on Figure 2.1.

The EC is inspired by biology, and therefore, technical terms from biology, genetics, and evolution are used [85]. It simulates mechanisms from evolution, where processes affecting the optimization are reproduction, mutation, competition, and selection. The reproduction is important for transfer of the genetic information from parents to their offspring. The transfer is influenced by defects, i.e. mutations, which can improve, or worsen the new generation. The competition is important in a constrained space, where each organism from the population competes with others to survive in the environment. Due to the competition, the selection is applied before new reproduction. The fittest organisms will survive and reproduce the offspring, whereas the less successful will perish [10].

2.1.2 Swarm intelligence

The swarm intelligence (SI) belongs to the group of evolutionary computation technique. Part of the SI method is particle swarm optimization (PSO) along with ant colony optimization [13], glowworm swarm optimization [77], or artificial bee colony algorithm [71].

The SI follows five basic principles [95]:

- principle of *proximity* - the population should be able to make elementary computations in time and space,
- principle of *quality* - the population should be able to respond to a quality factors of the environment,
- principle of *diverse response* - the population should not delimit its activations along narrow lines,
- principle of *stability* - the population should not change its behaviour every time, when the environment has changed,
- principle of *adaptability* - the population should be able to change its behaviour, if it has a computational value.

In the SI, each individual of a social community (e.g. ant, termite, bee, fish, bird, etc.) is ordinary, but as a unit they are able to accomplish a complicated task due to mutual cooperation [14]. The behaviour of organisms follows three simple rules [119]:

- *separation* - to avoid an overcrowding and collision (Fig. 2.2a),
- *cohesion* - to stay close to the neighbours (Fig. 2.2b),
- *alignment* - to match the direction and magnitude of velocity vector with the neighbours (Fig. 2.2c).

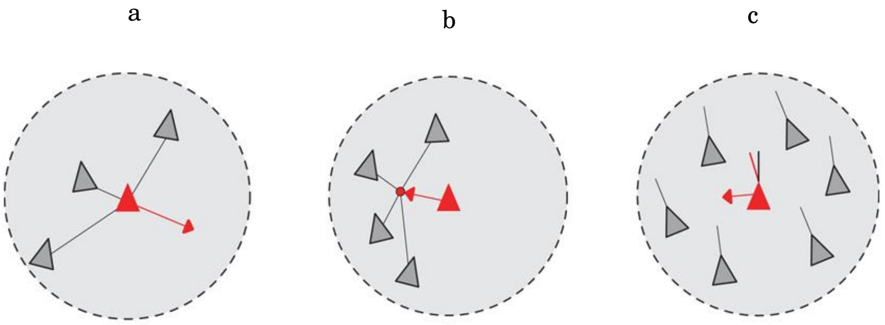

Figure 2.2: Rules of behaviour in SI, a) separation, b) cohesion, c) alignment [90]

2.2 Original equations

Particle swarm optimization is a meta-heuristic, stochastic computational technique, which is inspired by successive and unpredictable fly of birds [73]. The method has only a few parameters to adjust, and it is relatively easy to implement and use. The main advantage is also the fact, that PSO does not need gradient information of the objective function during the iterative search [49, 93, 94].

Particle swarm optimization contains a population of particles $i = 1,...,S$, where S is total number of individuals. Particles represent a potential solution of the optimization problem, and every new generation of individuals is closer to the searched optimum. The problem space has dimension $d = 1,...,Dim$, where Dim is total number of parameters.

Each particle i has its own position $\mathbf{X_i} = (\mathbf{x_1^i}, \mathbf{x_2^i}, ..., \mathbf{x_{Dim}^i})$ in the space, and velocity $\mathbf{V_i} = (\mathbf{v_1^i}, \mathbf{v_2^i}, ..., \mathbf{v_{Dim}^i})$, which are stored in the memory. Each particle i also maintains its previous best position $\mathbf{P_i} = (\mathbf{p_1^i}, \mathbf{p_2^i}, ..., \mathbf{p_{Dim}^i})$, and the best position among all particles $\mathbf{G} = (\mathbf{g_1}, \mathbf{g_2}, ..., \mathbf{g_{Dim}})$ [27, 41, 73].

Before the optimization process starts, the population needs to be initialize in the search space. The initialization of particle's position is randomly distributed in the range of $[x_{min}, x_{max}]$, and it is calculated as

$$\mathbf{X} = x_{min} + (x_{max} - x_{min}) \cdot \mathbf{R}, \tag{2.3}$$

where x_{min} and x_{max} are boundaries of the search space, $\mathbf{R}$ is vector of random numbers uniformly distributed in the range of $[0, 1]$ with length equal to Dim.

Positions can be also initialized through Latin hypercube sampling (LHS), type of stratified Monte Carlo sampling [91]. The search space is partitioned into n intervals of equal probability of $1/n$, where n is equal to the population size S. LHS then randomly selects one value from each interval [150]. Due to this selection, particles are uniformly distributed in the problem space without any clusters.

The particle's velocity can be initialized to 0 since the starting positions are already randomized [10]. Alternatively, the initialization of particle's velocity can be randomly distributed in the range of $[v_{min}, v_{max}]$. In the first experiment, the v_{max} was set to 100 000, but better approach is to limit the maximum velocity to the x_{max} [40]. Some authors [28, 44] set the value of maximum velocity as $v_{max} = k \cdot x_{max}$, where $0.1 < k < 1$. Larger value of maximum velocity facilitates global exploration, whereas smaller value encourages local exploitation [42, 134].

The original PSO algorithm consists of two main equations. One equation is for computing particle's velocity

$$\mathbf{v_d^i}(t+1) = \mathbf{v_d^i}(t) + c_1 \cdot \mathbf{r_{1d}}(t) \cdot (\mathbf{p_d^i}(t) - \mathbf{x_d^i}(t)) + c_2 \cdot \mathbf{r_{2d}}(t) \cdot (\mathbf{g_d}(t) - \mathbf{x_d^i}(t)), \tag{2.4}$$

and the second equation calculates particle's position

$$\mathbf{x_d^i}(t+1) = \mathbf{x_d^i}(t) + \mathbf{v_d^i}(t+1), \tag{2.5}$$

where t is time step, $\mathbf{r_{1d}}$ and $\mathbf{r_{2d}}$ are members of vectors **R1** and **R2** of random numbers uniformly distributed in the range of $[0,1]$, respectively, c_1 and c_2 are acceleration constants predefined by the user.

The component with $\mathbf{P_i}$ in Eq. 2.4 is referred as *cognition part*, and it represents the individual experience. The component with **G** is called *social part*, and it tells us about the cooperation among particles within the population [11, 43, 109].

The update of particle's position according to the original equations is depicted on Figure 2.3. The pseudo code of PSO algorithm is in Chapters 3 and 4.

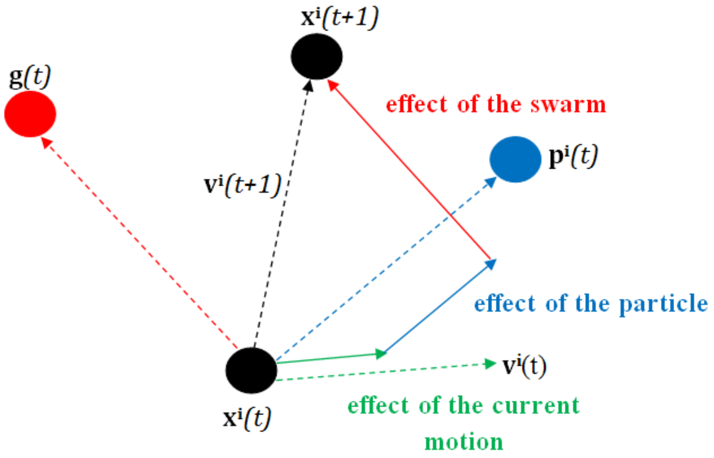

Figure 2.3: Updating of particle's position in PSO algorithm [57]

To improve the optimization ability of the original PSO algorithm, the method was variously modified. One approach for improving the PSO performance is adaptation of the topology of particles (Section 2.3). Another approaches are listed in Section 2.4.

2.3 Topology

Particles in the population interact and create connections due to transfer information about the best position achieved so far. These connections are called topology of the swarm. The set of individuals in connection is called the neighbourhood [25]. There are many types of topology [75, 76, 117], and the most commonly used are models called gbest and lbest.

The gbest model represents the method of global search of the problem space. Each particle is connected with all other particles (Fig. 2.4a). Hence, the information is transferred through the whole population [10]. The model has higher speed of convergence. It gives only one solution from the whole swarm called the best global particle. This individual acts as an attractant, it pulls other particles until the whole population converge to this location. The disadvantage of the gbest model is a possibility of premature convergence to the wrong particle [93].

The lbest model represents the method of local search. Each particle is connected with k individuals. The k is usually equal to 2, and the topology is a closed circle. This arrangement is called ring topology (Fig. 2.4b). The model tries to prevent the premature convergence by more attractants, one in every region of neighbourhood [81].

Kennedy and Mendes [75] discovered significant improvement using von Neumann topology. This structure is more densely connected than ring model, but less densely than gbest model. Each particle shares information with 4 other individuals (Fig. 2.4c).

Clerc [25] suggested random topology, where each particle is connected with k particles from the swarm, which are selected randomly. The k number is usually equal to 3 [26] (Fig. 2.4d).

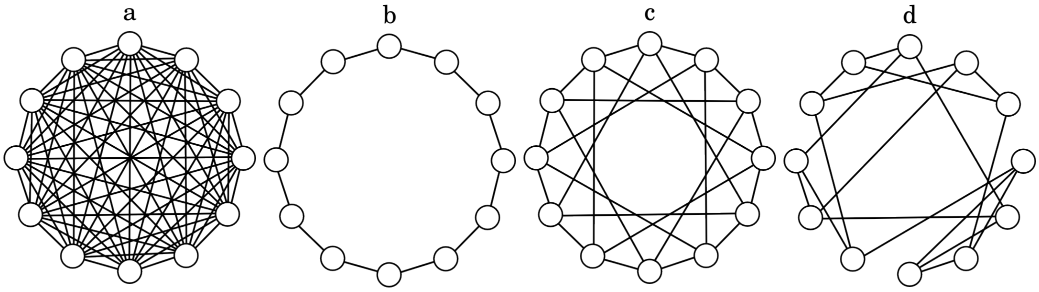

Figure 2.4: Graphical representation of different topologies, a) gbest, b) lbest, c) von Neumann, d) random

2.4 Modifications of PSO

In the optimization process, the premature convergence could appear, where the model could converge to the local optimum instead of the global one. Many researches were devoted avoiding this phenomenon [1, 111, 112].

The PSO method can be modified by different initialization of particles. The initialization can be made through low discrepancy sequences [135], quasi random sequence [110], or via opposite population, where the population with better fitness is selected for the PSO run [66].

Another approach is PSO algorithm with mutation operator. Wang et al. [140] proposed Cauchy mutation of the global best particle, Pant et al. [108] used adaptive mutation, or the power mutation operator was applied [65, 149].

The original PSO equation for calculating particle's velocity (Eq. 2.4) was modified to improve the optimization performance of the algorithm. Modifications summarized in this section are related to updating of the velocity from the previous time step, which is affected by a given parameter. The parameter is inertia weight (Section 2.4.1), or constriction factor (Section 2.4.2).

Other possibility for increasing the optimization ability is to use distributed version of the algorithm (Section 2.4.3). In this approach, the population is divided into several complexes, where the PSO algorithm runs at each complex individually.

All selected modifications, which are described in detail in the following text, were applied in experimental studies within this thesis. Chapter 3 compared chosen PSO variants on benchmark optimization problems, Chapter 4 and 5 applied them on real hydrological case studies.

2.4.1 Inertia weight

Inertia weight model was proposed for better control of the exploration and exploitation. The main goal was to create a modification of PSO, where is no need to choose v_{max}, but still ensure good convergence to the searched optimum. The first integration of the inertia weight into the PSO algorithm appeared in 1998 [43, 125, 126].

The particle's velocity is calculated as

$$\mathbf{v_d^i} = w \cdot \mathbf{v_d^i} + c_1 \cdot \mathbf{r_{1d}} \cdot (\mathbf{p_d^i} - \mathbf{x_d^i}) + c_2 \cdot \mathbf{r_{2d}} \cdot (\mathbf{g_d} - \mathbf{x_d^i}), \tag{2.6}$$

where w is inertia weight, and the other variables are the same as in Equation 2.4. The equation for calculating particle's position remains unchanged.

The acceleration constants are usually set to $c_1 = c_2 = 2$ [73]. The value of w can be different. In general, larger value encourages global search of the space (i.e. exploration), whereas smaller value encourages local search (i.e. exploitation) [127]. For $w > 1$, the velocity increases in time, and the particle can reach the border of the search space. For $w < 0$, the velocity decreases in time [14].

Modifications of PSO based on inertia weight can be divided into three classes [102]:

- constant and random inertia weight,
- time varying inertia weight,
- adaptive inertia weight,

and the following text is concerned with each of the group.

2.4.1.1 Constant and random inertia weight

In this class, the inertia weight is constant during the optimization, or it is determined randomly. This approach does not need any input data for setting the w value [102].

The constant inertia weight is shown in Equation 2.7. Bansal et al. [5] in their work set the constant $c = 0.7$. Consider

$$w = c. \tag{2.7}$$

Random inertia weight is calculated as

$$w = 0.5 + \frac{r}{2}, \tag{2.8}$$

where r is random number from the range $[0, 1]$, and thus, w is a number from the interval $[0.5, 1]$ [102]. Random inertia weight enables tracking optima in dynamic environment during the PSO algorithm run [42].

2.4.1.2 Time varying inertia weight

Time varying inertia weight belongs to the group of PSO modifications, which are defined as a function of time, or a function of a number of iterations. These methods are linear, or non-linear, the w value can be increasing, or decreasing.

Commonly used linear decline of inertia weight was proposed by Shi and Eberhart [127]. The w value linearly declines from the initial value w_{max} to the final value w_{min} as

$$w(iter) = \frac{iter_{max} - iter}{iter_{max}} \cdot (w_{max} - w_{min}) + w_{min}, \tag{2.9}$$

where $iter$ is current number of iteration, $iter_{max}$ is maximum number of iterations, $w_{max} = 0.9$, and $w_{min} = 0.4$.

Following equations describe chaotic models of calculating inertia weight. Equation 2.10 is called chaotic model, and Equation 2.11 is called chaotic random model. The variable z is computed as $z = 4 \cdot z \cdot (1 - z)$, where the initial value of z is chosen randomly from the range $[0,1]$. Variable r is random number from the interval $[0,1]$ [45]. Consider

$$w(iter) = (w_{max} - w_{min}) \cdot \frac{iter_{max} - iter}{iter_{max}} + w_{min} \cdot z, \tag{2.10}$$

$$w(iter) = 0.5 \cdot r(iter) + 0.5 \cdot z. \tag{2.11}$$

Two different non-linear time varying inertia weights are shown in following equations, where $iter$ is current number of iteration, w_{ini} is initial inertia weight randomly selected from the interval $[0,1]$, and u is constant from interval $[1.0001, 1.005]$, often equal to 1.0002 [102]. The equations are

$$w(iter) = w_{ini} \cdot u^{iter}, \tag{2.12}$$

$$w(iter) = \left(\frac{2}{iter}\right)^{0.3}. \tag{2.13}$$

2.4.1.3 Adaptive inertia weight

In the adaptive strategy of inertia weight, there is one or more parameters, which updates the w value through feedback information.

Panigrahi et al. [107] proposed adaptive inertia weight, where each particle has different w value according to the rank of that particle. Arumugam et al. [3] used a ratio of $\mathbf{g_d}$ and average value of $\mathbf{p_d^i}$ for determining the inertia weight at each iteration.

Nickabadi et al. [102] proposed adaptive inertia weight with one feedback parameter. The S parameter determines the success of each i^{th} particle at each number of iteration $iter$. The P parameter determines success of the swarm at each number of iteration based on the success of each particle, where n is size of the swarm. The corresponding equations are

$$S^i(iter) = \begin{cases} 1 & \text{if } f(\mathbf{p_d^i}(iter)) < f(\mathbf{p_d^i}(iter-1)) \\ 0 & \text{if } f(\mathbf{p_d^i}(iter)) = f(\mathbf{p_d^i}(iter-1)) \end{cases}, \tag{2.14}$$

$$P(iter) = \frac{\sum_{i=1}^{n} S^i(iter)}{n}. \tag{2.15}$$

The inertia weight is then $w(iter) = f(P(iter))$. When using linear function in this relationship, w is calculated as

$$w(iter) = (w_{max} - w_{min}) \cdot P(iter) + w_{min}, \tag{2.16}$$

where the range of $[w_{min}, w_{max}]$ is $[0,1]$.

2.4.2 Constriction factor

Constriction factor is a parameter, which was first implemented into PSO algorithm by Clerc in 1999 [24]. It is sometimes called Clerc factor, or Clerc constriction factor. Its incorporation into the velocity equation increases the convergence of the algorithm [40]. The updated equation is

$$\mathbf{v_d^i} = K \cdot (\mathbf{v_d^i} + c_1 \cdot \mathbf{r_{1d}} \cdot (\mathbf{p_d^i} - \mathbf{x_d^i}) + c_2 \cdot \mathbf{r_{2d}} \cdot (\mathbf{g_d} - \mathbf{x_d^i})), \tag{2.17}$$

where K is constriction factor and the other variables are the same as in Equation 2.4. The equation for calculation particle's position remains the same.

The value of constriction factor is calculated by

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi}|}, \tag{2.18}$$

where φ is positive constant, for which $\varphi = c_1 + c_2$, and $\varphi > 4$ [24, 27].

Acceleration constants are usually $c_1 = c_2 = 2.05$, and thus, $\varphi = 4.1$, and $K = 0.7298$ [24, 40]. Putting these values into Equation 2.17 leads to

$$\mathbf{v_d^i} = 0.7298 \cdot (\mathbf{v_d^i} + 2.05 \cdot \mathbf{r_{1d}} \cdot (\mathbf{p_d^i} - \mathbf{x_d^i}) + 2.05 \cdot \mathbf{r_{2d}} \cdot (\mathbf{g_d} - \mathbf{x_d^i})). \tag{2.19}$$

After adjustment of Equation 2.19, we get Equation 2.20, which is equivalent to PSO equation using parameter of inertia weight (Eq. 2.6) with parameters equal to $w = 0.7298$, and $c_1 = c_2 = 1.4962$ [40]. Therefore,

$$\mathbf{v_d^i} = 0.7298 \cdot \mathbf{v_d^i} + 1.4962 \cdot \mathbf{r_{1d}} \cdot (\mathbf{p_d^i} - \mathbf{x_d^i}) + 1.4962 \cdot \mathbf{r_{2d}} \cdot (\mathbf{g_d} - \mathbf{x_d^i}). \tag{2.20}$$

Eberhart and Shi [40] compared in their work optimization PSO with inertia weight and with constriction factor. In all tested benchmark functions, the modification with constriction factor gave better results, and the speed of convergence was higher. Further improvement was achieved with the condition when $v_{max} = x_{max}$.

2.4.3 Distributed version

Distributed version of optimization algorithm is an important strategy for improving the optimization performance. This approach does not influence the velocity equation, it influences the swarm of particles itself.

Duan el al. [34] suggested the method called Shuffled complex evolution (SCE), where the population is divided into several complexes, and the optimization algorithm runs at each complex individually. After the termination criteria is met, particles return to the swarm, and the shuffling and redistributing to sub-swarms take place. The SCE method is robust, effective and efficient for many optimization problems. It was also included to the PSO algorithm as a method SCE-PSO [152], or DMS-PSO (Dynamic multi-swarm PSO) [80].

In the SCE-PSO method, the population S is divided into p complexes, where each complex contains m particles. From each complex is chosen q particles according to the best achieved fitness values. The selected particles create a sub-swarm. At each sub-swarm runs the PSO algorithm until the maximum number of iteration T is achieved. After that, all particles return to the swarm, the population is shuffled, particles are sorted based on their fitness values, and the swarm is again divided into p complexes. The whole procedure is repeated until the termination criteria is met. Each step of the SCE-PSO algorithm is displayed on Figure 2.5, and is more explained in the original paper of Yan et al. [152].

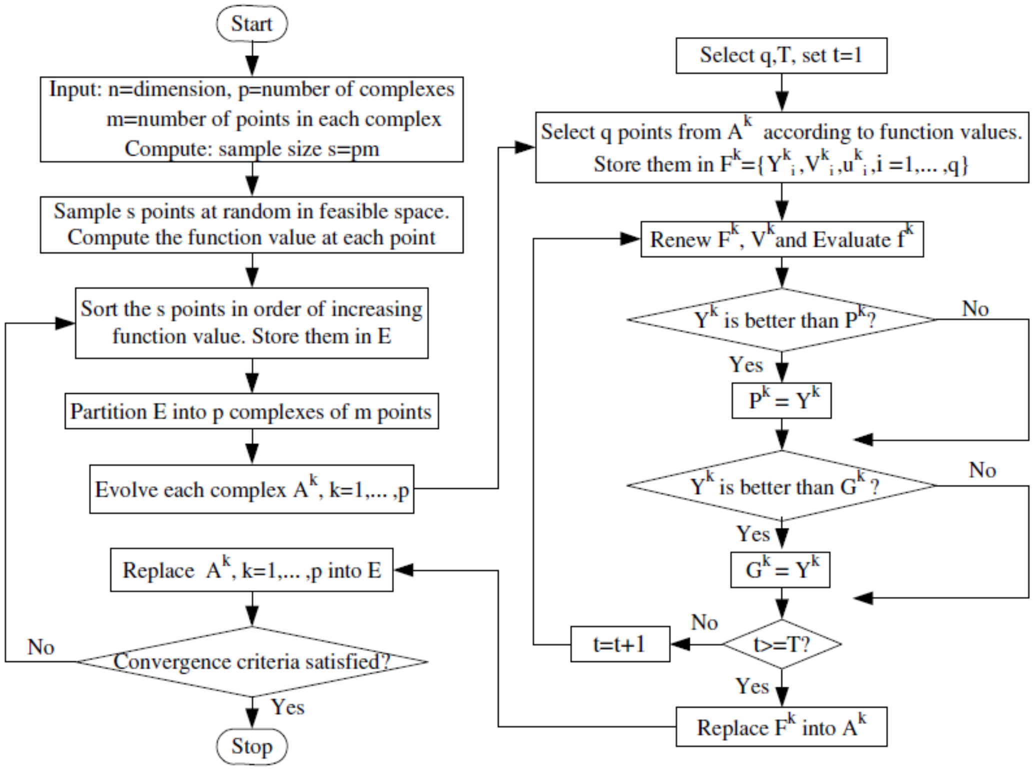
Start
Input: n=dimension, p=number of complexes
m=number of points in each complex
Compute: sample size s=pm
Sample s points at random in feasible space.
Compute the function value at each point
Sort the s points in order of increasing
function value. Store them in E
Partition E into p complexes of m points
Evolve each complex A^k, k=1,... ,p
Replace A^k, k=1,... ,p into E
No
Convergence criteria satisfied?
Yes
Stop
Select q,T, set t=1
Select q points from A^k according to function values.
Store them in F^k={Y^k_i,V^k_i,u^k_i,i =1,... ,q}
Renew F^k, $V^k$and Evaluate f^k
Y^k is better than P^k?
No
Yes
P^k = Y^k
Y^k is better than G^k ?
No
Yes
G^k = Y^k
No
t=t+1
t>=T?
Yes
Replace F^k into A^k

Figure 2.5: Scheme of algorithm SCE-PSO [152]

The difference in DMS-PSO method is that the swarm is dynamic, and the size of population is small. The whole population is divided into many small complexes, which are often shuffled based on various procedures [80]. Compared to the SCE-PSO algorithm, in the DMS-PSO is the PSO optimization used for all particles from the swarm, not only for particles with the best fitness value.

2.5 Objective functions

During optimization, the main aim is to find an optimal value of an objective function f. In this thesis, the optimization is a minimization. The process of searching the minimal value depends on the used optimization method, but the goal remains the same.

This section describes benchmark problems and hydrological indexes, which are all objective functions used in this doctoral thesis. For testing and comparison purposes, the benchmark problems are solved. Chapter 3 contains details on comparison of chosen PSO modifications on 11 benchmark problems. Part of the research in Chapter 4 also deals with benchmark functions due to testing new proposed PSO variant.

Optimization based on hydrological indexes is commonly used in practical experiments within the field of hydrological modelling. Second part of Chapter 4 summarizes results of optimization of hydrological model Bilan, and Chapter 5 displays training of artificial neural network with PSO algorithm. For both studies, chosen hydrological indexes were optimized.

2.5.1 Benchmark problems

Benchmark problems serve for comparing different optimization techniques, or for testing new proposed optimization method. Benchmark functions are precisely defined, the user knows their formula, range of the search space, and the position of the optimal value. Results of finding the optimal value are comparable across different research for all scientists.

The benchmark objective functions used in this doctoral thesis were prepared for the special session on real-parameter single objective optimization of Congress on Evolutionary Computation 2005 [131]. The analysed uni-modal functions, which have only one local optimum, are Sphere, Schwefel 1.2, Elliptic rotated, Schwefel 1.2 with noise, and Schwefel 2.6 function (Fig. 2.6, $f_1 - f_5$). From multimodal problems, which have more local optima, I analysed Rosenbrock, Griewank rotated, Ackley rotated, Rastrigin, Rastrigin rotated, and Weierstrass rotated function (Fig. 2.6, $f_6 - f_{11}$). All functions have shifted global optimum.

The summary of selected benchmark problems is in Table 2.1. The range defines the range of the searched space with the dimension Dim, and $f(\mathbf{X_{opt}})$ shows the functional value of the shifted global optimum. For formulas of the benchmark problems, see Table 3.2 in the next chapter, or Suganthan et al. [131].

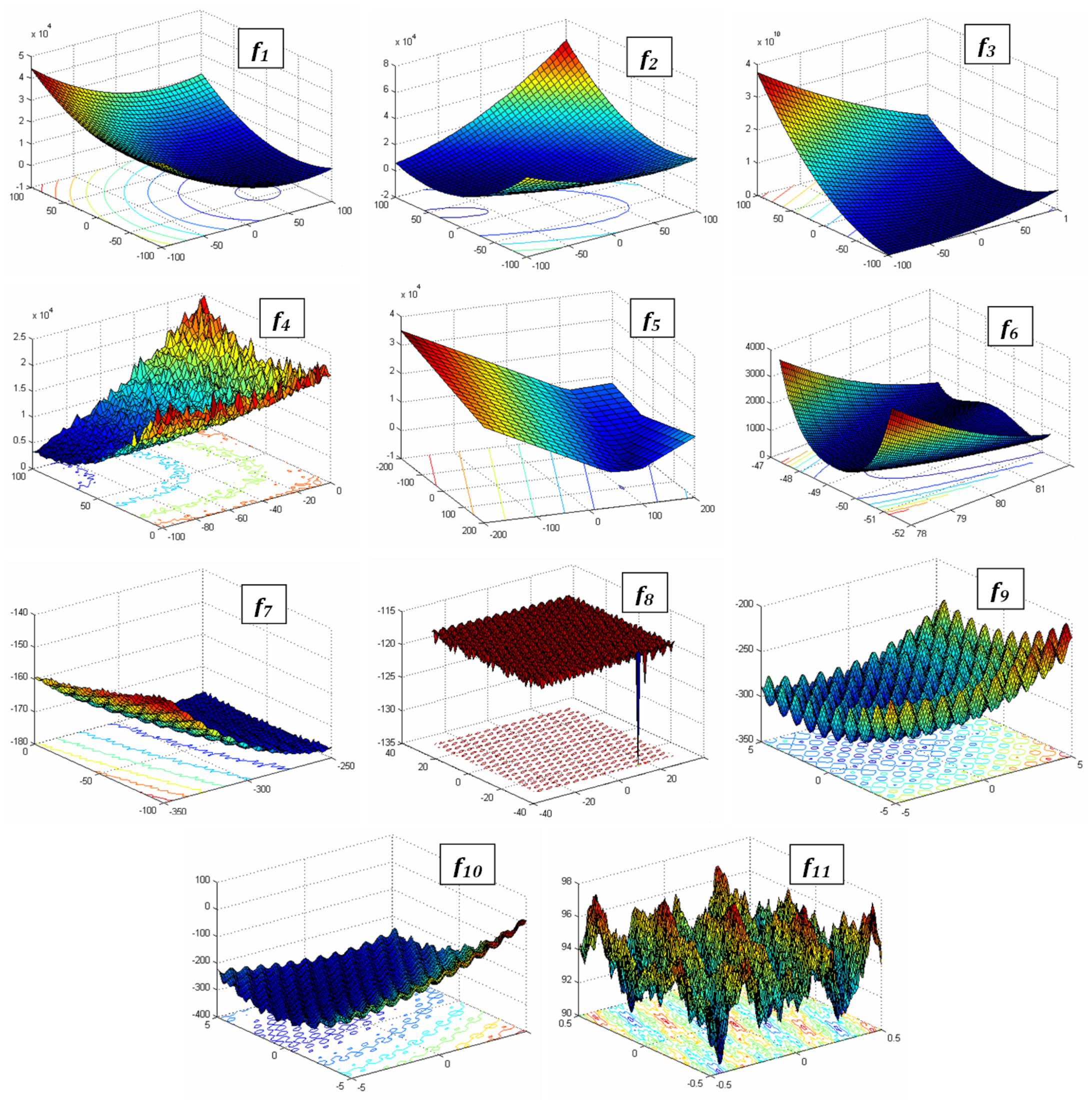

Figure 2.6: Selected uni-modal ($f_1 - f_5$), and multi-modal ($f_6 - f_{11}$) benchmark functions [131]

2.5.2 Hydrological indexes

Optimization methods in hydrological modelling are used for calibration of models, estimation of rainfall-runoff relationships, meteorological forecasts, or runoff predictions [83, 148, 158]. Hydrological index serves as an objective function, and it also determines the quality of hydrological model.

Dawson et al. [29] divided objective criteria widely used in hydrology into three classes:

- statistical parameters of observed and modelled time series,
- statistical parameters of the residual error,
- dimensionless coefficients.

Table 2.1: Selected benchmark functions

	Function	**Range**	$f(\mathbf{X_{opt}})$
f_1	Sphere	$[-100,100]^{Dim}$	-450
f_2	Schwefel 1.2	$[-100,100]^{Dim}$	-450
f_3	Elliptic rotated	$[-100,100]^{Dim}$	-450
f_4	Schwefel 1.2 noise	$[-100,100]^{Dim}$	-450
f_5	Schwefel 2.6	$[-100,100]^{Dim}$	-310
f_6	Rosenbrock	$[-100,100]^{Dim}$	390
f_7	Griewank rotated	$[0,600]^{Dim}$	-180
f_8	Ackley rotated	$[-32,32]^{Dim}$	-140
f_9	Rastrigin	$[-5,5]^{Dim}$	-330
f_{10}	Rastrigin rotated	$[-5,5]^{Dim}$	-330
f_{11}	Weierstrass rotated	$[-0.5,0.5]^{Dim}$	90

The group of statistical parameters of observed and modelled time series is characterized using standard descriptive statistics. The main magnitudes are minimum, maximum, mean, variance, standard deviation, or lag-one autocorrelation coefficient [22, 29]. This group will not be used for optimization within this thesis, because the relationship between observed and modelled values needs to be taking into account.

In the class of statistical parameters of the residual error, the residuals present difference between observed and simulated value. The group is divided into two types of error. If the criteria provide quantitative assessment of the model in the same units as the variables of interest, they are called absolute errors. To this category belong absolute maximum error (AME), mean error (ME), mean absolute error (MAE), mean squared error (MSE), root mean squared error ($RMSE$), etc. If the statistical parameters of the residual error are not dependent on the units, they are called relative errors, which are expressed in percentage or ratios. The category of relative errors includes mean relative error (MRE), mean squared relative error ($MSRE$), relative volume error (RVE), etc.

The dimensionless coefficients determine model performance, and they are often used for optimization in hydrological modelling. Within this group belong Nash-Sutcliffe coefficient (R^2), coefficient of persistence (PI), etc.

The summary of selected objective functions frequently used in hydrological modelling is listed in Table 2.2, where OBS is observed variable, MOD is modelled value, $\overline{OBS}$ is average of observed values, and n is total number of observations. The summation in the equations is in terms of temporal coordinates. The spatial coordinates are not used within this thesis, because the outflow

Table 2.2: Summary of selected objective functions frequently used in hydrological modelling

Criterion	Equation	Range	The best
AME	$max(\|OBS_i - MOD_i\|)$	$[0;+\infty]$	0
ME	$\frac{1}{n}\sum_{i=1}^{n}(OBS_i - MOD_i)$	$[-\infty;+\infty]$	0
MAE	$\frac{1}{n}\sum_{i=1}^{n}(\|OBS_i - MOD_i\|)$	$[0;+\infty]$	0
MSE	$\frac{1}{n}\sum_{i=1}^{n}(OBS_i - MOD_i)^2$	$[0;+\infty]$	0
$RMSE$	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(OBS_i - MOD_i)^2}$	$[0;+\infty]$	0
MRE	$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{OBS_i - MOD_i}{OBS_i}\right)$	$[-\infty;+\infty]$	0
$MSRE$	$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{OBS_i - MOD_i}{OBS_i}\right)^2$	$[0;+\infty]$	0
RVE	$\frac{\sum_{i=1}^{n}(OBS_i - MOD_i)}{\sum_{i=1}^{n}(OBS_i)}$	$[-\infty;+\infty]$	0
R^2	$1 - \frac{\sum_{i=1}^{n}(OBS_i - MOD_i)^2}{\sum_{i=1}^{n}(OBS_i - \overline{OBS})^2}$	$[-\infty;1]$	1
PI	$1 - \frac{\sum_{i=1}^{n}(OBS_i - MOD_i)^2}{\sum_{i=1}^{n}(OBS_i - OBS_{i-1})^2}$	$[-\infty;1]$	1

from the catchment at specific location is analysed (Chapter 4), or drought index characteristic for the whole catchment is simulated (Chapter 5).

2.6 Hydrological case studies

The main topic of this doctoral thesis is application of particle swarm optimization in hydrological modelling. Therefore, this section provides several recent hydrological case studies with utilization of PSO.

Jiang et al. [70] applied PSO for calibration of the rainfall-runoff model HIMS. They compared classical PSO algorithm with PSO methods using complexes and shuffling, and they found out that PSO with sub-swarms are significantly better than the original one.

Zambrano-Bigiarini and Rojas [154] proposed R package called hydroPSO, which serves to calibrate hydrological models. They compared six PSO algorithms on ten objective functions, and concluded that hydroPSO is better for nine functions. The hydroPSO was afterwards applied for calibration of hydrological model SWAT and groundwater flow model MODFLOW. The package showed good optimization ability, and it was effective and efficient for both surface and groundwater flow.

Calibration of groundwater flow was also solved by Gaur et al. [48]. They applied PSO in the catchment of the river Dore in France for estimating two hydraulic problems: 1) maximum amount of water pumping from a groundwater body, and 2) minimum costs of a new pumping system. They found out that the PSO is suitable for determining the optimal placing of wells, and also for determining the optimal volume of pumped water.

Lü et al. [84] applied PSO algorithm for estimation of soil water content in root zone. Their results show that PSO was suitable for this type of research, if the data from the topsoil were available. The model was also effective when there is no information about hydraulic conductivity of the soil.

PSO has also important use in training of the artificial neural networks. The optimization method was used to determine the value of the weights, which minimalized the error of the model. This approach was very simple, therefore, the PSO algorithm replaced the algorithm of the back-propagation [20, 74, 93].

Chau [21] proposed a model with multilayer perceptron trained by particle swarm optimization. He applied the model in the catchment of the Shing Mun River for estimation of water level. The results of his study show that the model achieved higher accuracy in shorter time compared to the method of back-propagation.

Good results were also obtained by Senthil Babu and Vinayagam [124], who combined PSO with ANN for prediction of the roughness of the surface. The forecast of precipitation with ANN trained by PSO was also sufficient and accurate [59, 147]. Combination of ANN models with PSO was also an effective tool for forecasting of drought indices, and thus, it was suitable for prevention of the drought event [8, 32, 63].

Chapter 3

A comparison of selected modifications of the particle swarm optimization algorithm

Abstract

In this chapter, 27 modifications of the original particle swarm optimization (PSO) algorithm are compared. The analysis evaluated nine basic PSO types, which differ according to the swarm evolution as controlled by various inertia weights and constriction factor. Each of the basic PSO modifications was analysed using three different distributed strategies. In the first strategy, the entire swarm population is considered as one unit (OC-PSO), the second strategy periodically partitions the population into equally large complexes according to the particle's functional value (SCE-PSO), and the final strategy periodically splits the swarm population into complexes using random permutation (SCERand-PSO).

All variants are tested using 11 benchmark functions that were prepared for the special session on real-parameter optimization of CEC 2005. It was found that the best modification of the PSO algorithm is a variant with adaptive inertia weight. The best distribution strategy is SCE-PSO, which gives better results than do OC-PSO and SCERand-PSO for seven functions. The Sphere function

This chapter is based on the publication: Jakubcová M., Máca P., and Pech P., 2014: A comparison of selected modifications of the particle swarm optimization algorithm. *Journal of Applied Mathematics*, vol. 2014, Article ID 293087, 10 pp, doi: 10.1155/2014/293087.

showed no significant difference between SCE-PSO and SCERand-PSO. It follows that a shuffling mechanism improves the optimization process.

3.1 Introduction

Particle swarm optimization (PSO) is a stochastic, meta-heuristic computational technique for searching the optimal regions from multidimensional space. It is an optimization method inspired by social behaviour of organisms, and was established by Kennedy and Eberhart in 1995 [73]. The technique is based on iterative work with a population. PSO is an evolutionary computation (EC) method within the group of techniques known as swarm intelligence (SI) [31, 143]. PSO mimics the movement of flock of birds or school of fish using simple rules for adjusting the particle location, which is adjusted by means of its velocity information.

PSO's main benefits are that there are few parameters to adjust, and the method is easy to implement. Another advantage of PSO over derivative based local search methods is that there is no need for the gradient information during the iterative search when solving complicated optimization problems [49, 93, 94].

While it has been successfully applied to solve many test and real-life optimization problems [4, 50, 100], the PSO method often suffers from premature convergence, and, as a result, from the optimization process's finding a merely local optimum. In order to achieve better algorithm performance, the original PSO algorithm has been modified by adding the parameter inertia weight or constriction factor [18, 102, 125, 127].

Another important strategy for improving EC algorithms relies on division of the original population into sub-swarms or complexes which simultaneously search across the parametric space and exchange information according to some prescribed rule. Periodic shuffling is a typical example [34, 100, 138].

In order to explore the interaction of modifications in particle velocities together with the different types of distributed PSO versions, this chapter analyzes 27 different PSO variants. Nine modifications of the PSO algorithm, in which the original particle velocities are altered using different approaches for setting the inertia weights and constriction factor [102], are combined with three strategies for swarm distribution. The population is either considered as one unit (OC-PSO), or the swarm is divided into several complexes, either according to the functional value (SCE-PSO) [34], or randomly (SCERand-PSO).

The remainder of this chapter is organized as follows. Section 3.2 describes the particle swarm optimization method. The original equations and modifications of PSO algorithm are included. Section 3.3 describes different strategies for distribution of PSO. The experiment and obtained results are compared in Section 3.4. Conclusions are discussed in Section 3.5.

3.2 Particle swarm optimization

Particle swarm optimization is a global optimization method applied to find the optimal solution $\mathbf{X^{opt}}$ of objective function f. The sought optimum is most generally a minimum value. There exists a population of particles $i = 1,..,S$, where S is the total number of individuals. All particles search through the problem space of dimension $d = 1,..,Dim$, where Dim is the total number of dimensions. Each particle stores information about its position and velocity. The vector of the i^{th} particle's position is $\mathbf{X_i} = (\mathbf{x^i_1}, \mathbf{x^i_2}, ..., \mathbf{x^i_{Dim}})$, the vector of the i^{th} particle's velocity is $\mathbf{V_i} = (\mathbf{v^i_1}, \mathbf{v^i_2}, ..., \mathbf{v^i_{Dim}})$. Each particle maintains a memory of its previous best position which is represented as $\mathbf{P_i} = (\mathbf{p^i_1}, \mathbf{p^i_2}, ..., \mathbf{p^i_{Dim}})$. The best position among all particles from the swarm is represented as $\mathbf{G} = (\mathbf{g_1}, \mathbf{g_1}, ..., \mathbf{g_{Dim}})$. Equations (3.1) and (3.2) are the original PSO equations for computing a new velocity and new position. Consider

$$\mathbf{v^i_d}(t+1) = \mathbf{v^i_d}(t) + c_1 \cdot r_1 \cdot (\mathbf{p^i_d}(t) - \mathbf{x^i_d}(t)) + c_2 \cdot r_2 \cdot (\mathbf{g_d}(t) - \mathbf{x^i_d}(t)), \tag{3.1}$$

$$\mathbf{x^i_d}(t+1) = \mathbf{x^i_d}(t) + \mathbf{v^i_d}(t+1), \tag{3.2}$$

for all $i \in 1..S$, $d \in 1..Dim$, where t is a time step, c_1 and c_2 are acceleration constants predefined by the user, r_1 and r_2 are random numbers uniformly distributed in $[0,1]$. The component with $\mathbf{P_i}$ in Eq. (3.1) is known as the cognition part, and it tells us about the individual experience of the particle. The component with $\mathbf{G}$ is called the social part, and it represents the cooperation among particles within the swarm [109].

The simplified original PSO algorithm is shown in Algorithm 1. Initialization of a particle's position is randomly distributed in the range of $[x_{min}, x_{max}]$ as shown in

$$\mathbf{X} = x_{min} + (x_{max} - x_{min}) \cdot rand(), \tag{3.3}$$

where $rand()$ is a random number uniformly distributed in $[0,1]$, while x_{min} and x_{max} are boundaries of the search space, and their values depend on the benchmark function [57]. In this chapter, initialization of particles is through Latin hypercube sampling (LHS).

Algorithm 1 Original PSO algorithm

```
initialize the position and velocity of all particles
repeat
    for each particle $i = 1$ to $S$ do
        if $(f(\mathbf{X_i}) < f(\mathbf{P_i}))$ then
            $\mathbf{P_i} = \mathbf{X_i}$
        end if
        $\mathbf{G} = min\{\mathbf{P_0}, \mathbf{P_1}, ..., \mathbf{P_S}\}$
        for each dimension $d = 1$ to $Dim$ do
            $\mathbf{v_d^i} = \mathbf{v_d^i} + c_1 \cdot r_1 \cdot (\mathbf{p_d^i} - \mathbf{x_d^i}) + c_2 \cdot r_2 \cdot (\mathbf{g_d} - \mathbf{x_d^i})$
            $\mathbf{x_d^i} = \mathbf{x_d^i} + \mathbf{v_d^i}$
        end for
    end for
until termination criteria is met
```

Particle's initial velocity could be randomly distributed in the range of $[-v_{max}, v_{max}]$, or alternatively, the velocities could be initialized to 0, since the starting positions are already randomized [10]. In initial experiments, the value of v_{max} was set to 100 000, in subsequent experiments and applications it was found, that a better approach is to limit v_{max} to x_{max} [40]. Other authors [28, 44] have set the value of maximum velocity as $v_{max} = k \cdot x_{max}$, where $0.1 < k < 1$. A larger value of v_{max} facilitates global exploration, whereas a smaller value of v_{max} encourages local exploitation [41]. In this chapter, $v_{max} = x_{max}$ was applied for initial particle velocity.

3.2.1 Modifications of the PSO algorithm

The original PSO equation was modified to improve the ability for optimization. The first group of modifications consists in incorporating the parameter of *inertia weight* w, the second in using the parameter of *constriction factor* K. In the present study, nine variants of PSO algorithm were used and tested (Tab. 3.1), including eight modifications using w and one modification with K.

The use of the inertia weight parameter (Eq. (3.4)) was developed by Shi and Eberhart [127], and it has provided for improved performance.

$$\mathbf{v_d^i} = w \cdot \mathbf{v_d^i} + c_1 \cdot r_1 \cdot (\mathbf{p_d^i} - \mathbf{x_d^i}) + c_2 \cdot r_2 \cdot (\mathbf{g_d} - \mathbf{x_d^i}). \tag{3.4}$$

There are many methods of computing the inertia weight value. Nickabadi et al. [102] divided those techniques into three classes which are applied in this chapter: 1) constant and random inertia weight, 2) time varying inertia weight strategies, and 3) adaptive inertia weight. They compared all modifications

Table 3.1: Summary of PSO modifications

Label	**Equation**
AdaptW	$w = (w_{max} - w_{min}) \cdot P_s + w_{min}$
ChaoticRandW	$w(iter) = 0.5 \cdot rand() + 0.5 \cdot z$
ChaoticW	$w(iter) = (w_{max} - w_{min}) \cdot \frac{iter_{max} - iter}{iter_{max}} + w_{min} \cdot z$
ConstantW	$w = c$
ConstrFactor	$K = \frac{2}{\vert 2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi} \vert}$
LinTimeVaryingW	$w(iter) = \frac{iter_{max} - iter}{iter_{max}} \cdot (w_{max} - w_{min}) + w_{min}$
NonlinTimeConstW	$w(iter) = w_{ini} \cdot u^{iter}$
NonlinTimeW	$w(iter) = \left(\frac{2}{iter} \right)^{0.3}$
RandomW	$w = 0.5 + \frac{rand()}{2}$

employing benchmark functions, and proposed a PSO algorithm using adaptive inertia weight.

The constant (“ConstantW”) and random (“RandomW”) inertia weights are used, where no input information is required. Bansal et al. [5] discussed their work with Shi and Eberhart [125], and set the constant inertia weight equal to 0.7. Gimmler et al. [51] proposed using constant inertia weight for hybrid particle swarm optimization. The best constant w was 0.2. The constant inertia weight for the “ConstantW” modification was set to 0.7 for this study, because the experiment is more similar to that of Bansal et al. [5]. Eberhart and Shi [42] had proposed random inertia weight, where w is a variable with uniform distribution within $[0.5, 1]$.

Time varying inertia weight is defined as a function of time or number of iterations, and this method may be linear or nonlinear. In linear decreasing w (“LinTimeVaryingW”) developed by Shi and Kennedy [127], inertia weight decreases linearly from $w_{max} = 0.9$ to $w_{min} = 0.4$. This method of determining the inertia weight value is very common [151, 153]. Eberhart and Shi [40] compared linearly decreasing w with constriction factor, and found that better performance was achieved when constriction factor was used. The chaotic model (“ChaoticW”) and chaotic random model (“ChaoticRandW”) of inertia weight were proposed by Feng et al. [45], where $z = 4 \cdot z \cdot (1 - z)$, and the initial value of z is uniformly distributed in $[0, 1]$. Two modifications of nonlinear time varying inertia weight are used. The “NonlinTimeW” and the “NonlinTimeConstW”, where parameter u is set to 1.0002, w_{ini} is the initial value for inertia weight uniformly distributed in $[0, 1]$ and $iter$ is the actual number of functional evaluations [102].

One modification of adaptive inertia weight proposed by Nickabadi et al. [102] is used ("AdaptW"), because it had demonstrated the best performance in the original paper. The w value is adapted based on one feedback parameter. The value S shows the success of particles, and is defined as in Eq. (3.5), P_s is the success percentage of the swarm, and it is computed as in Eq. (3.6), where n is the size of the population. The range of inertia weights $[w_{min}, w_{max}]$ is $[0,1]$. Consider

$$S(i,t) = \begin{cases} 1 & \text{if} \quad fit(\mathbf{p_d^i}(t)) < fit(\mathbf{p_d^i}(t-1)) \\ 0 & \text{if} \quad fit(\mathbf{p_d^i}(t)) = fit(\mathbf{p_d^i}(t-1)) \end{cases}, \tag{3.5}$$

$$P_s(t) = \frac{\sum_{i=1}^{n} S(i,t)}{n}. \tag{3.6}$$

Beyond variants using inertia weight, the next modification of the original PSO algorithm consists in incorporating the parameter of constriction factor (Eq. (3.7) and (3.8)). This strategy was first used by Clerc [24], and it increases convergence of the algorithm. The method is called "ConstrFactor". Another approach to constriction factor is that of Bui et al. [18]. They proposed a time-dependent strategy, where they used nonlinear decay rules to adapt K. Their results are not better than those obtained when using the setting in accordance with Clerc [24], and therefore, the K value was calculated using

$$\mathbf{v_d^i} = K \cdot (\mathbf{v_d^i} + c_1 \cdot r_1 \cdot (\mathbf{p_d^i} - \mathbf{x_d^i}) + c_2 \cdot r_2 \cdot (\mathbf{g_d} - \mathbf{x_d^i})), \tag{3.7}$$

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi}|}, \tag{3.8}$$

where $\varphi = c_1 + c_2$, and $\varphi > 4$.

3.3 Distribution of PSO

All nine modifications are used with three strategies of swarm distribution (SD). Changes in behaviour of the population for each modification and strategy were observed. The first distributed strategy considered the whole population as one unit called it OC-PSO. In the next SD, the population was divided into several complexes according to the particle's functional value (SCE-PSO), or through random permutation (SCERand-PSO).

3.3.1 One complex strategy (OC-PSO)

In the OC-PSO method, the entire population is considered as a single unit. All particles participate in the PSO algorithm and share information about the best position achieved so far.

3.3.2 Shuffled complex evolution (SCE-PSO)

During the optimization process a premature convergence to a local optimum could appear instead of the global optimum. Many researchers in this field are devoted avoiding this premature convergence [1, 111, 112]. To address this, Duan et al. [34] proposed shuffled complex evolution (SCE). Yan et al. [152] combined SCE with the particle swarm optimization algorithm (SCE-PSO).

The SCE-PSO method is described simply below, and is shown in Algorithm 2. After the first initialization, the entire population is divided into sub-swarms according to the functional values of the individuals. All particles are sorted in increasing order, and than each i^{th} complex receives the parent individuals $\mathbf{X_i}$,$\mathbf{X_{i+NC}}$,$\mathbf{X_{i+2NC}}$,..., where NC is the number of complexes [34, 89]. The PSO algorithm is applied at each complex. After running a predefined number of iterations in all complexes, all particles return to the swarm, and the shuffling and redistribution to complexes according to the functional value are again made. This is repeated until the termination criteria are satisfied.

The shuffling mechanism preserves the population diversity and helps to prevent premature convergence. For this study, the shuffling was performed after the running of a predefined number of generations in each complex [89]. Another approach allows the shuffling to occur randomly with some associated probability [142].

In the original SCE-PSO method [152], only a predefined number of particles from each complex is chosen to participate in the PSO algorithm. In this chapter, the number of participating individuals is equal to the number of particles in the complex. This means, that all particles from the complex are inputs to the PSO algorithm.

3.3.3 Random shuffled complex evolution (SCERand-PSO)

Random shuffled complex evolution differs from SCE-PSO in that the entire population is divided into complexes according to random permutation. There is

Algorithm 2 Algorithm SCE-PSO

```
Require: S, NC, N_comp , max_eval , fitness_lim
 1: initialize population $\mathbf{X}$
 2: while (number_eval ≤ max_eval) || (fitness_best ≥ fitness_lim) do
 3:     $\mathbf{E} \leftarrow$ sorted $\mathbf{X}$ in increasing order according to the functional value
 4:     for $i = 1$ to NC do
 5:         $\mathbf{A_i} \leftarrow$ divided $\mathbf{E}$ into NC complexes
 6:         run PSO
 7:         for $j = 1$ to N_comp do
 8:             if $(f(\mathbf{X_j}) < f(\mathbf{P_j}))$ then
 9:                 $\mathbf{P_j} = \mathbf{X_j}$
10:             end if
11:             if $(f(\mathbf{X_j}) < f(\mathbf{G}))$ then
12:                 $\mathbf{G} = \mathbf{X_j}$
13:             end if
14:         end for
15:     end for
16: end while
```

no sorting by functional value. The algorithm for computing random permutation is according to Durstenfeld [38].

Algorithm 2 is applied for the SCERand-PSO, except that at line 3 the following substitution is made: **E** ← sorted **X** according to random permutation.

3.4 Experiment and results

3.4.1 Experimental setup

After running several tests with different parameter settings, the following setup was found to be the best.

The position of individuals is initialized randomly between lower and upper bounds of the search space through Latin hypercube sampling (LHS). The range of the problem space depends on the benchmark function (Tab. 3.2). LHS is a type of stratified Monte Carlo sampling first described by McKay, Beckman and Conover in 1979 for the analysis of output from a computer code [91]. The range, which is in PSO optimization defined by lower and upper bounds of the search space, is portioned into n intervals of equal probability $1/n$. The n value is in PSO equal to the population size. LHS then randomly selects one value from each interval [150]. Due to this selection, particles are uniformly distributed in the search space.

Table 3.2: Summary of benchmark functions

	Function	**Range**	$f(\mathbf{X^{opt}})$
Sphere	$f_1(x) = \sum_{i=1}^{d} z_i^2$	$[-100,100]^d$	−450
Schwefel 1.2	$f_2(x) = \sum_{i=1}^{d} (\sum_{j=1}^{i} z_j)^2$	$[-100,100]^d$	−450
Elliptic rotated	$f_3(x) = \sum_{i=1}^{d} ((10^6)^{\frac{i-1}{d-1}} z_i^2)$	$[-100,100]^d$	−450
Schwefel 1.2 noise	$f_4(x) = (\sum_{i=1}^{d} (\sum_{j=1}^{i} z_j)^2) \cdot (1 + 0.4\lvert N(0,1)\rvert)$	$[-100,100]^d$	−450
Schwefel 2.6	$f_5(x) = max\{\lvert A_i x - B_i\rvert\}$	$[-100,100]^d$	−310
Rosenbrock	$f_6(x) = \sum_{i=1}^{d-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$	$[-100,100]^d$	390
Griewank rotated	$f_7(x) = \sum_{i=1}^{d} \frac{z_i^2}{4000} - \prod_{i=1}^{d} cos(\frac{z_i}{\sqrt{i}}) + 1$	$[0,600]^d$	−180
Ackley rotated	$f_8(x) = -20exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} z_i^2}) - exp(\frac{1}{d}\sum_{i=1}^{d} cos(2\pi z_i)) + 20 + e$	$[-32,32]^d$	−140
Rastrigin	$f_9(x) = \sum_{i=1}^{d} (z_i^2 - 10cos(2\pi z_i) + 10)$	$[-5,5]^d$	−330
Rastrigin rotated	$f_{10}(x) = \sum_{i=1}^{d} (z_i^2 - 10cos(2\pi z_i) + 10)$	$[-5,5]^d$	−330
Weierstrass rotated	$f_{11}(x) = \sum_{i=1}^{d} [\sum_{k=0}^{20} (0.5^k cos(2\pi 3^k (z_i + 0.5)))] - d\sum_{k=0}^{20} (0.5^k cos(2\pi 3^k \cdot 0.5))$	$[-0.5,0.5]^d$	90

In accordance with Eberhart and Shi [40], the maximum value of velocity is set to x_{max}. The value of acceleration constants c_1 and c_2 in variants with inertia weight is set to 2. In modifications with constriction factor, the K value is set to 0.729 and $c_1 = c_2 = 1.49445$ [40].

According to Eberhart and Shi [43], the population size is set to 25. In the OC-PSO method, all particles are solved together. In the SCE-PSO and SCERand-PSO methods, individuals are uniformly divided into 6 complexes, where each complex contains 25 particles. The number of shuffling is set to 5. The maximum number of function evaluations is $10\,000 \cdot Dim$, and the dimension of the solution is set to 30. For analyzing the results, the total number of optimization runs is set to 25. Each run stops when the maximum number of evaluations is achieved.

3.4.2 Benchmark problems

For comparison purposes, 11 benchmark functions prepared for the special session on real-parameter optimization of CEC 2005 [131] were used. All functions have shifted global optima, some of them is rotated or with noise. The benchmark functions are summarized in Table 3.2. The aim is to find the minimum of all functions.

The optimization problem is constrained except for function f_7. Particles move only in restricted space and cannot cross the boundaries. This means, that each position of particle i is bounded by lower and upper limits [19]. In the PSO algorithm, it is reflected such that the particles must lie within the range $[x_{min}, x_{max}]$. If a new particle position is outside the boundaries, that particle retains the position of its parent. In function f_7, the global optimum is situated outside the range [131], and therefore, the optimization problem is unconstrained, and particles can cross the boundaries.

There exist two versions of the PSO algorithm: *global* and *local*. In the global variant, the neighborhood consists of all particles of the swarm. In the local variant, each particle is assigned to a neighborhood consisting of a predefined number of particles [19, 113]. For the OC-PSO method, the global variant is used. For the SCE-PSO and SCERand-PSO, the local variant is used, and particles share the information about their best positions only with other particles from a given complex.

3.4.3 Results and discussion

The non-parametric Wilcoxon test was used for statistical comparison. Inputs to those calculations were the best fitness values achieved for all modifications. The null hypothesis H_0 of the Wilcoxon test is that differences between algorithms have a median of zero. The H_0 is rejected if the p-value is less than 0.05 [39, 47]. Algorithms were written in C++, and computations of p-value and graphs were made in the program R.

Table 3.3 reflects the best modification and distributed strategy for each function. In the table, the minimum, 25% quartile, median, 75% quartile, maximum, mean and standard deviation are indicated. All 25 program runs of each modification and strategy were compared in each numbered evaluation. The values in Table 3.3 reflect the best fitness achieved, and report other statistical indices belonging to the same numbered evaluation. As can be seen, strategy SCE-PSO produced the best solution in seven functions. Strategy SCERand-PSO produced the best solution in two functions (f_9,f_{10}), and in one function (f_5), the best solution was from strategy OC-PSO. For function f_1, there was no significant difference between strategy SCE-PSO and SCERand-PSO.

Upon closer examination and as seen in Table 3.4, “AdaptW” and “NonlinTimeConstW” are the best modifications for unimodal functions (f_1 - f_5). The poorest variants are “ConstantW” and “ConstrFactor”. The best PSO modification for multimodal functions (f_6 - f_{11}) is “AdaptW”, and the poorest is “ConstantW”.

For rotated functions (f_3, f_7, f_8, f_{10}, f_{11}), the best modification of the PSO algorithm appears to be “AdaptW”, and the poorest is “CostantW”. For functions where there is no transformation matrix to rotate them is the best variant “AdaptW”, and the poorest are “ConstantW” and “ConstrFactor”.

It is clear that the best modification of the particle swarm optimization algorithm for the selected benchmark functions is “AdaptW”, i.e. adaptive inertia weight. The variant called “NonlinTimeConstW” also produced good results. On the other hand, the poorest modifications appear to be “ConstantW” and “ConstrFactor”.

The convergence to the global optimum using particle swarm optimization is good, but only in three of the eleven benchmark functions is the obtained error value less than 10^{-8}. In spite of this, the shuffling mechanism improves the optimization. Strategies SCE-PSO and SCERand-PSO are better than OC-PSO in ten functions.

The global minimum was achieved in three benchmark functions - f_1, f_2 and f_7,

Table 3.3: Statistical indices of the best solutions of 11 benchmark functions

func.	**modification**	**SD** [a]	**min**	*25%*	**median**	*75%*	**max**	**mean**	**std**
f_1	AdaptW	2	0.00E+00	5.68E-14	1.14E-13	1.71E-13	2.27E-13	1.02E-13	6.14E-14
	AdaptW	3	0.00E+00	5.68E-14	1.14E-13	1.14E-13	5.59E-05	4.74E-06	1.53E-05
f_2	AdaptW	2	5.68E-14	2.27E-13	3.98E-13	6.82E-13	1.53E-12	5.16E-13	4.33E-13
f_3	AdaptW	2	4.17E+05	7.08E+05	1.02E+06	1.57E+06	2.30E+06	1.15E+06	5.78E+05
f_4	NonlinTimeConstW	2	7.09E-02	7.56E-01	2.21E+00	4.38E+00	1.94E+03	1.20E+02	4.27E+02
f_5	NonlinTimeConstW	1	1.31E+03	2.78E+03	3.21E+03	4.57E+03	6.14E+03	3.66E+03	1.26E+03
f_6	AdaptW	2	1.71E-03	3.83E+00	4.06E+00	7.47E+00	1.80E+01	5.76E+00	4.87E+00
f_7	NonlinTimeConstW	2	2.84E-14	9.86E-03	1.23E-02	2.22E-02	6.87E-02	1.70E-02	1.73E-02
f_8	AdaptW	2	2.06E+01	2.14E+01	2.16E+01	2.18E+01	2.20E+01	2.15E+01	3.89E-01
f_9	ChaoticW	3	2.89E+01	6.78E+01	8.51E+01	1.12E+02	6.07E+02	1.21E+02	1.27E+02
f_{10}	ChaoticRandW	3	7.87E+01	1.63E+02	4.22E+02	4.85E+02	6.56E+02	3.44E+02	1.88E+02
	LinTimeVaryingW	3	8.38E+01	2.03E+02	4.02E+02	4.81E+02	5.71E+02	3.54E+02	1.54E+02
f_{11}	LinTimeVaryingW	2	2.66E+01	3.12E+01	3.25E+01	3.52E+01	4.78E+01	3.38E+01	4.97E+00

[a] SD = swarm distribution, where 1 is for OC-PSO, 2 is for SCE-PSO, 3 is for SCERand-PSO.

Table 3.4: Estimation of the best and poorest modifications

f_1	**Best**	**Poorest**
OC-PSO	AdaptW, ChaoticRandW, ChaoticW, NonlinTimeConstW	ConstantW
SCE-PSO	AdaptW	ConstantW
SCERand-PSO	AdaptW, NonlinTimeConstW	ConstantW
f_2	**Best**	**Poorest**
OC-PSO	AdaptW, LinTimeVaryingW	ConstrFactor
SCE-PSO	AdaptW	ConstrFactor
SCERand-PSO	AdaptW	ConstrFactor
f_3	**Best**	**Poorest**
OC-PSO	LinTimeVaryingW	ConstantW
SCE-PSO	AdaptW, NonlinTimeW	ConstantW
SCERand-PSO	AdaptW, NonlinTimeW	ConstantW, RandomW
f_4	**Best**	**Poorest**
OC-PSO	ChaoticRandW, LinTimeVaryingW	ConstrFactor
SCE-PSO	NonlinTimeConstW	ConstrFactor
SCERand-PSO	NonlinTimeConstW	ConstrFactor
f_5	**Best**	**Poorest**
OC-PSO	ChaoticRandW, NonlinTimeConstW	ConstantW
SCE-PSO	ChaoticRandW, ChaoticW, NonlinTimeConstW	ConstantW
SCERand-PSO	ChaoticRandW, ChaoticW, LinTimeVaryingW, NonlinTimeConstW	ConstantW
f_6	**Best**	**Poorest**
OC-PSO	ChaoticRandW, NonlinTimeConstW	ConstantW
SCE-PSO	AdaptW	ConstantW
SCERand-PSO	AdaptW	ConstantW
f_7	**Best**	**Poorest**
OC-PSO	NonlinTimeConstW	ConstantW, RandomW
SCE-PSO	NonlinTimeConstW	ConstantW, ConstrFactor
SCERand-PSO	AdaptW, NonlinTimeW	ConstantW, ConstrFactor, LinTimeVarying
f_8	**Best**	**Poorest**
OC-PSO	AdaptW	ConstrFactor
SCE-PSO	AdaptW	LinTimeVarying
SCERand-PSO	AdaptW	ConstantW, ConstrFactor, LinTimeVaryingW, NonlinTimeConstW

Table 3.4: Continued

f_9	Best	Poorest
OC-PSO	ChaoticW	ConstantW, ConstrFactor, RandomW
SCE-PSO	ChaoticW	ConstantW, ConstrFactor
SCERand-PSO	ChaoticW	ConstantW
f_{10}	**Best**	**Poorest**
OC-PSO	ChaoticRandW	ConstantW
SCE-PSO	ChaoticW	ConstantW
SCERand-PSO	ChaoticRandW, ChaoticW, LinTimeVaryingW	ConstantW
f_{11}	**Best**	**Poorest**
OC-PSO	AdaptW, LinTimevaryingW, NonlinTimeConstW, NonlinTimeW	ConstantW, ConstrFactor
SCE-PSO	LinTimevaryingW	ConstantW, ConstrFactor, RandomW
SCERand-PSO	AdaptW	ConstantW, ConstrFactor, RandomW

and the results are comparable with those of Hansen [55], who compares eleven optimization algorithms on twelve functions. The particle swarm optimization algorithm achieved the global minimum in functions f_1, f_2, f_3, f_6 and f_7. Obtained results using SCE-PSO are better than those of the optimization algorithms BLX-MA and DE, and are as good as those from CoEVO as reported by Hansen [55].

Bui et al. [18] achieved the global optimum in two unimodal functions f_1 and f_2 using the PSO method APSO1, and in three unimodal functions f_1, f_2 and f_4 using the DPSO method. APSO is PSO algorithm with adaptive constriction factor, and in DPSO, the bound of the velocity is adapted. None of their algorithms converge to the global minimum in multimodal functions [18]. In this regard, results obtained in this chapter are better.

Nickabadi et al. [102] compared six inertia weight adjusting methods on 15 test problems with dimension set to 30. If looking only at functions solved for this study, Nickabadi et al. [102] achieved the global optimum in functions f_1, f_2 and f_8. They obtained the best results using the adaptive inertia weight, which they had proposed (modification “AdaptW” in this chapter). Their results are comparable with those achieved in the present study. The difference being that all functions analysed in this chapter have shifted global optimum, whereas in Nickabadi et al. [102] did not.

Figure 3.1 presents the convergence graphs for each function while utilizing the best modification of the PSO algorithm. The x-axis indicates the number of function evaluations, and the y-axis the logarithmic value of the best fitness, which is the difference between the searched and the best achieved functional value. A decline with the number of evaluations is clearly visible for all functions, thus indicating the approach to the global optimum.

3.5 Conclusions

This chapter compared 27 variants of particle swarm optimization algorithm. Eight modifications were performed using the parameter inertia weight and one modification using constriction factor. Both parameters improved the optimization. All modifications were tested with three strategies of swarm distribution, which were in terms of population. The population was either considered as a single unit (OC-PSO), or it was divided into several complexes. Division into complexes was made according to the functional value of each particle (SCE-PSO), or through random permutation (SCERand-PSO).

The main aim of this study was to find the global minima of eleven benchmark functions prepared for the special session on real-parameter optimization of CEC 2005. The achievement of the minimum is when the obtained error value is less than 10^{-8}. The global minimum was obtained in two unimodal functions (f_1 and f_2), and in one multimodal function (f_7). The original particle swarm optimization has slow convergence to the global optimum, but the shuffling mechanism improves the optimization.

The best modification of the PSO algorithm is the variant called “AdaptW”. The best choice for selected benchmark functions is to use the parameter of inertia weight, where the w value is adapted based on a feedback parameter. The best strategy for swarm distribution is SCE-PSO. Shuffled complex evolution particle swarm optimization with allocation of particles into complexes according to their functional values is better than OC-PSO and SCERand-PSO.

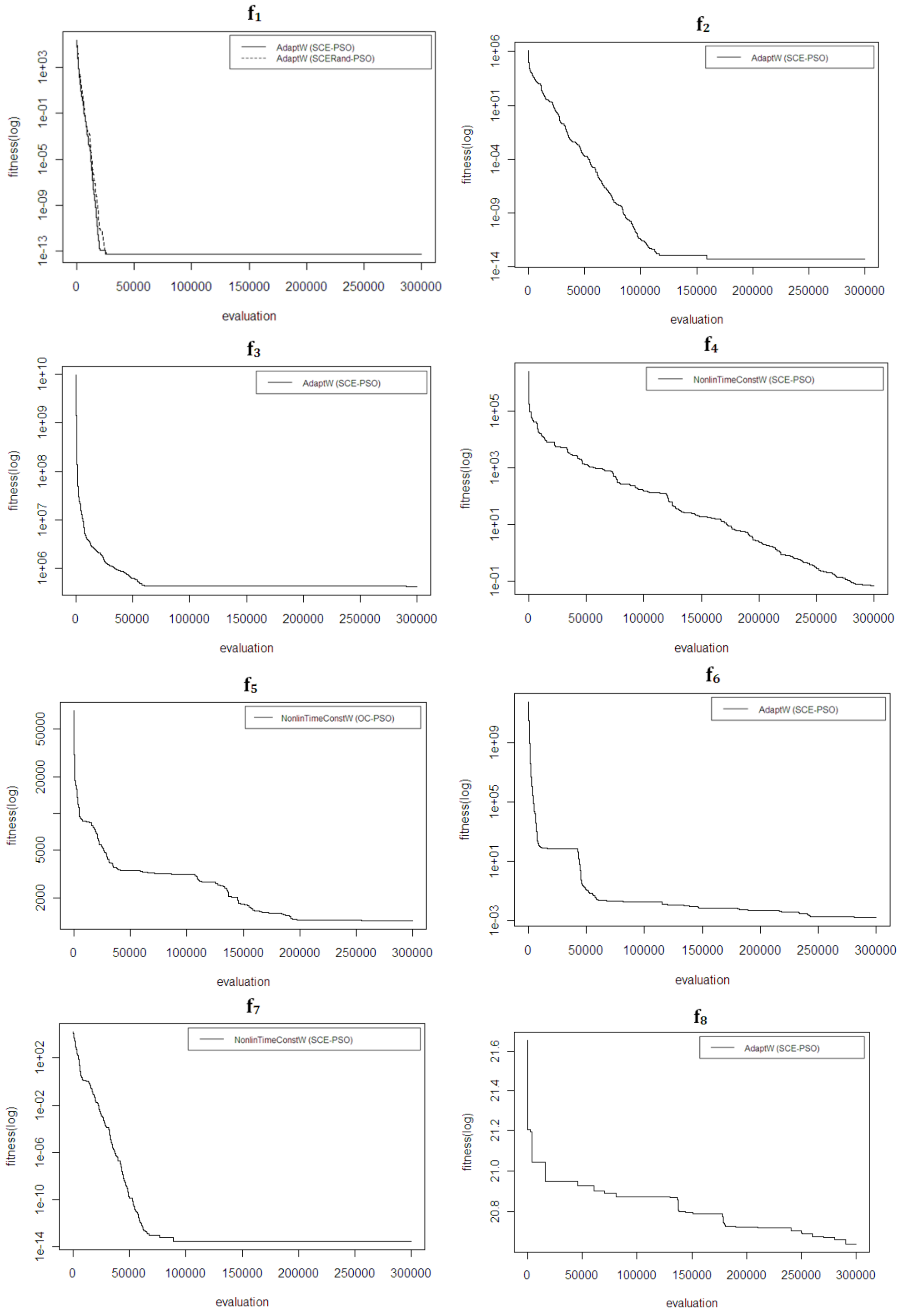

Figure 3.1: Convergence graphs for best fitness of the best modifications and strategy of PSO for functions f_1-f_{11} according to Table 3.3

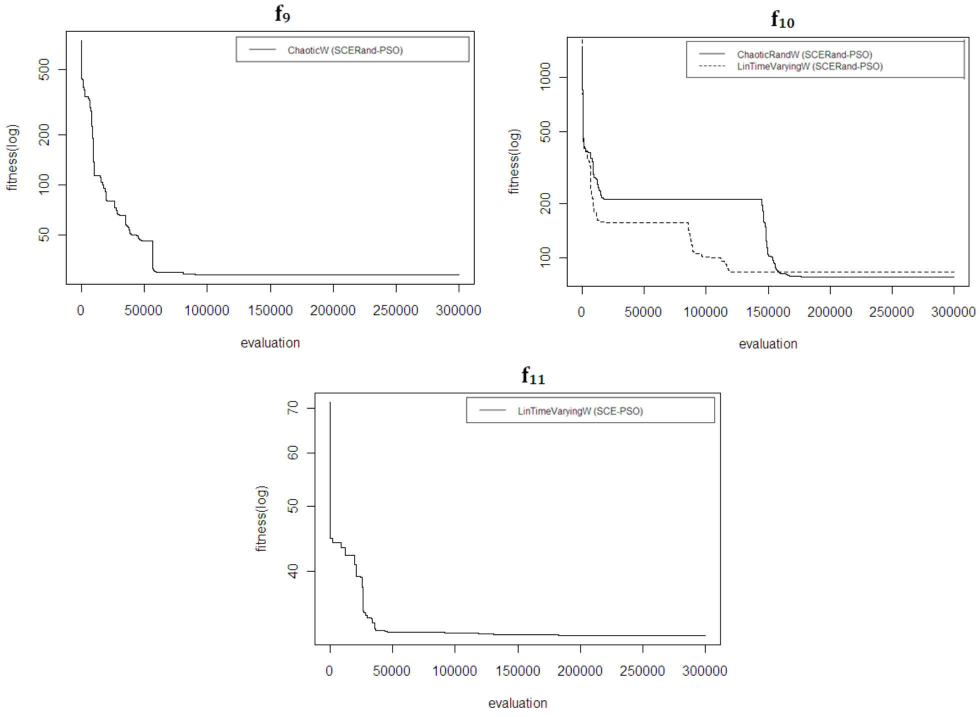

Figure 3.1: Continued

Chapter 4

Parameter estimation in rainfall-runoff modelling using distributed versions of particle swarm optimization algorithm

Abstract

The presented chapter provides the analysis of selected versions of the particle swarm optimization (PSO) algorithm. The tested versions of the PSO were combined with the shuffling mechanism, which splits the model population into complexes, and performs distributed PSO optimization. One of them is a new proposed PSO modification - APartW, which enhances the global exploration and local exploitation in the parametric space during the optimization process through the new updating mechanism applied on the PSO inertia weight. The performances of four selected PSO methods were tested on 11 benchmark optimization problems, which were prepared for the special session on single-objective real-parameter optimization CEC 2005. The results confirm, that the tested new APartW PSO variant is comparable with other existing distributed PSO versions - AdaptW and LinTimeVarW. The distributed PSO versions were developed for

This chapter is based on the publication: Jakubcová M., Máca P., and Pech P., 2015: Parameter estimation in rainfall-runoff modelling using distributed versions of particle swarm optimization algorithm. *Mathematical Problems in Engineering*, vol. 2015, Article ID 968067, 13 pp, doi: 10.1155/2015/968067.

finding the solution of inverse problems related to the estimation of parameters of hydrological model Bilan. The results of the case study, made on the selected set of 30 catchments obtained from MOPEX database, show, that tested distributed PSO versions provide suitable estimates of Bilan model parameters, and thus can be used for solving related inverse problems during the calibration process of studied water balance hydrological model.

4.1 Introduction

Particle swarm optimization (PSO) was established in 1995 by Kennedy and Eberhart [73]. It is an evolutionary optimization technique inspired by a behaviour of population of individuals, which form groups or swarms like flock of birds, or school of fishes. It works with population of particles, which are finding the optimal solution during their search within the search space by collaboration between individuals, and by exchanging information about their best position in the space.

The PSO belongs between the stochastic, meta-heuristic evolutionary computational techniques, which are based on the swarm intelligence [31]. The main benefits of this optimization are small number of parameters, which need to be adjusted, and an easy implementation. When comparing with standard local search methods, the PSO does not require a knowledge about the gradient of the optimized function [49, 94].

Its optimization process often suffers from premature convergence, or from trapping in the local optimum. Therefore, the recent PSO research focuses on the development of new adaptation strategies. The most important adaptations improve the particle's velocity estimates by the adaptation of PSO inertia weight [5, 45].

Special attention is also put on the development of distributed version of PSO [12, 103]. The original population is divided into sub-swarms, or complexes. The complexes are simultaneously evolving over the parametric space, and they periodically exchange information according to some prescribed migration rules [34, 138].

The PSO has been successfully applied into many real-life optimization problems in engineering. In recent years, the PSO optimization significantly enhances the estimation of parameters of hydrological models [4, 48, 50, 84].

For example, Jiang et al. [70] applied PSO for calibration of the rainfall-runoff model HIMS. They compared classical PSO algorithm with distributed PSO versions, which are using the complexes and shuffling mechanism. They found out that the distributed PSO variants are significantly better than the original one.

Zambrano-Bigiarini and Rojas [154] developed the stand alone global optimization for calibrating the hydrological models based on the PSO (called hydroPSO and available for R interpreter). The methods enable to test different PSO versions on calibration of analysed hydrological model. The group of tested PSO versions showed good optimization performance, and it was an effective and efficient tool for calibration of both surface and groundwater hydrological models.

The main aim of this chapter is to test the selected PSO distributed versions on single-objective benchmark optimization problems, and to apply them on calibration of hydrological model Bilan. The case study is conducted on 30 US catchments, for which the data of hydrological and meteorological forcings are obtained from MOPEX experiment [35].

The rest of the chapter is organized as follows. Section 4.2 provides details of the standard PSO algorithm and its tested modifications. Section 4.3 explains methodology used during single-objective benchmark optimization problems and methodology of the rainfall-runoff model simulations with description of the Bilan rainfall-runoff model. Results are summarized in Section 4.4 and discussion is provided in Section 4.5. Finally, the main findings are concluded in Section 4.6.

4.2 Particle swarm optimization

This section provides description of tested versions of distributed PSO. At first, the original PSO is described, and then the analysed variants of PSO are provided. The distribution strategy is explained in the last subsection.

4.2.1 The original PSO

The original PSO algorithm estimates a new particle's location using information of particle's velocity. The velocity $\overrightarrow{V}_i = (v_{i1}, v_{i2}, ..., v_{iDim})$ is updated as

$$\overrightarrow{V}_i^{t+1} = \overrightarrow{V}_i^t + c_1 \cdot \overrightarrow{U}_1^t \otimes \left(\overrightarrow{P}_i^t - \overrightarrow{X}_i^t\right) + c_2 \cdot \overrightarrow{U}_2^t \otimes \left(\overrightarrow{G}^t - \overrightarrow{X}_i^t\right), \tag{4.1}$$

and the location $\overrightarrow{X}_i = (x_{i1}, x_{i2}, ..., x_{iDim})$ is simply defined as

$$\overrightarrow{X}_i^{t+1} = \overrightarrow{X}_i^t + \overrightarrow{V}_i^{t+1}, \tag{4.2}$$

for all $i = 1, ..., S$, where S is total number of particles in swarm population, $d = 1, ..., Dim$, where Dim is total number of dimensions (i.e. the number of parameters of hydrological model), c_1 and c_2 denotes acceleration constants predefined by the user, $\overrightarrow{U}_1$ and $\overrightarrow{U}_2$ are independent random vectors sampled from a uniform distribution in the range $[0,1]$.

The component with particle's previous best position $\overrightarrow{P}_i = (p_{i1}, p_{i2}, ..., p_{iDim})$ in Equation (4.1) represents the cognition knowledge of swarm particles. When optimization problem is the minimization, the $f(\overrightarrow{P}_i^t) \leq f(\overrightarrow{X}_i^t)$ for all so far known locations of given particle $\overrightarrow{X}_i^t$. The component with the best position among all particles $\overrightarrow{G} = (g_1, g_2, ..., g_{Dim})$ in Equation (4.1) controls the social influence of swarm particles. For all $\overrightarrow{X}_i$ in the population is the $f(\overrightarrow{G}^t) \leq f(\overrightarrow{X}_i^t)$. The f is the analysed single-objective function [73, 128].

The particle's locations are in this chapter initialized randomly within the search space using the initialization based on the Latin hypercube sampling technique [91, 150]. Particle's velocity is initialized randomly from the interval $[-\overrightarrow{V}^{max}, \overrightarrow{V}^{max}]$, where the $\overrightarrow{V}^{max}$ is equal to $\overrightarrow{X}^{max}$ [10, 40].

4.2.2 Analysed modifications of PSO

Four versions of the PSO algorithm were analysed. They differ according to applied particle's velocity adaptation. All versions were tested as asynchronous distributed PSO. The modifications are following.

ConstrFactor. In the first modification, the parameter of constriction factor K is implemented into the PSO algorithm [24]. If the adapted particle's velocity with K is as

$$\overrightarrow{V}_i^{t+1} = K \cdot \left(\overrightarrow{V}_i^t + c_1 \cdot \overrightarrow{U}_1^t \otimes \left(\overrightarrow{P}_i^t - \overrightarrow{X}_i^t\right) + c_2 \cdot \overrightarrow{U}_2^t \otimes \left(\overrightarrow{G}^t - \overrightarrow{X}_i^t\right)\right), \tag{4.3}$$

then

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi}|}, \tag{4.4}$$

where $\varphi = c_1 + c_2$, and $\varphi > 4$.

LinTimeVarW. In the second PSO modification, the linearly decreasing inertia weight w is used [127]. The adapted particle's velocity equation is shown in

$$\overrightarrow{V}_i^{t+1} = w \cdot \overrightarrow{V}_i^t + c_1 \cdot \overrightarrow{U}_1^t \otimes \left(\overrightarrow{P}_i^t - \overrightarrow{X}_i^t\right) + c_2 \cdot \overrightarrow{U}_2^t \otimes \left(\overrightarrow{G}^t - \overrightarrow{X}_i^t\right). \tag{4.5}$$

The value of inertia weight decreases at each iteration linearly from $w_{max} = 0.9$ to $w_{min} = 0.4$, and is simply defined as

$$w_{iter} = \frac{iter_{max} - iter}{iter_{max}} \cdot (w_{max} - w_{min}) + w_{min}. \tag{4.6}$$

AdaptW. One of the adaptive inertia weight from [102] was used in this chapter. The modification of the inertia weight parameter is as

$$w_{iter} = (w_{max} - w_{min}) \cdot \overrightarrow{Ps}^t + w_{min}, \tag{4.7}$$

for which

$$\overrightarrow{Ps}^t = \frac{\sum_{i=1}^{n} \overrightarrow{S}_i^t}{n}, \tag{4.8}$$

$$\overrightarrow{S}_i^t = \begin{cases} 1 & \text{if} \quad f(\overrightarrow{P}_i^t) < f(\overrightarrow{P}_i^{t-1}) \\ 0 & \text{if} \quad f(\overrightarrow{P}_i^t) = f(\overrightarrow{P}_i^{t-1}) \end{cases}, \tag{4.9}$$

where $\overrightarrow{S}_i$ shows the success of i^{th} particle, $\overrightarrow{Ps}$ is the success percentage of the swarm, n is the size of the population, $w_{min} = 0$ and $w_{max} = 1$. The adaptive inertia weight value is updated based on a feedback parameter which is in this case the variable of percentage of success.

The AdaptW PSO method provided the best performance in the original paper of Nickabadi et al. [102], and it was also the best modification of inertia weight out of total 27 distributed variants of PSO (for details see [67]).

APartW. The proposed new adaptive inertia weight modification is based on the current position of the particle, and it combines the global exploration and local exploitation in the space. The value of the inertia weight parameter is updated at each generation according to the development of particle's location. If the particle improves its position compared to the best location achieved so far, it is closer to the searched optimal value. So, if $f(\overrightarrow{X}_i^t) < f(\overrightarrow{P}_i^{t-1})$, the local exploitation is supported. The inertia weight is calculated as

$$\overrightarrow{w}_i^t = \left(\frac{w_{max} + w_{min}}{2} - w_{min}\right) \cdot \overrightarrow{U}^t + w_{min}, \tag{4.10}$$

and thus, for random vector $\overrightarrow{U}$ sampled from a uniform distribution in the range $[0,1]$, $w_{min} = 0.1$ and $w_{max} = 0.9$, the resulted inertia weight is between $[0.1, 0.5]$.

On the other hand, if the particle does not achieve better position, the global exploration is encouraged. So, if $f(\overrightarrow{X}_i^t) \geq f(\overrightarrow{P}_i^{t-1})$, the resulted inertia weight is computed as

$$\overrightarrow{w}_i^t = \left(\frac{w_{max} + w_{min}}{2} - w_{min}\right) \cdot \overrightarrow{U}^t + \left(\frac{w_{max} + w_{min}}{2}\right), \tag{4.11}$$

where the variables are the same as in Eg. (4.10), and the inertia weight is thus between $[0.5, 0.9]$.

4.2.3 Distributed version of PSO

During the optimization process, a premature convergence to a local optimum could appear. Due to this, different distribution strategies of the swarm were developed [67, 156]. A new distribution strategy of the swarm called shuffled complex evolution (SCE) was proposed by [34]. In this chapter, the SCE-PSO method introduced by [152] is used, where the shuffled complex evolution is combined with the PSO optimization. The only difference in this research is that all particles from the complex are participating in the PSO algorithm, not only predefined number of them.

In the SCE strategy, after initialization of particle's position and velocity, all population is divided into predefined number of complexes NC. The division is according to the functional value of each particle. All particles are sorted in increasing order, and then each j^{th} complex receives $\overrightarrow{X}_j, \overrightarrow{X}_{j+NC}, \overrightarrow{X}_{j+2NC}, ...$ particles [89]. Each complex simultaneously searches through the parametric space, and after predefined number of iterations in one complex, all particles return to the swarm. The shuffling of particles and redistribution into the complexes are made, and the process is repeated until the termination criteria are satisfied.

The shuffling mechanism preserves the population diversity and helps to prevent premature convergence. The SCE-PSO was already applied for comparison of 27 modifications of PSO [67], and it was found that the SCE strategy gives significantly better results.

In this chapter, all four PSO variants were extended into a distributed version using SCE-PSO technique. Since the APartW algorithm is a new proposed version, the simplified PSO algorithm using APartW adaptation and SCE mechanism is shown in Algorithm 3, where S_comp is number of particles in one complex. The other PSO modifications differ only on Lines 9, 11 and 16 in Algorithm 3.

4.3 Methodology of single-objective optimization problems

Distributed versions of PSO were analysed and tested on two sets of single-objective optimization problems. All optimization problems were minimizations. The total number of analysed optimization problems was 15.

The first set is represented by the 11 benchmark problems, which were specially prepared for CEC 2005 single-objective optimization session [131]. The total

Algorithm 3 PSO algorithm with APartW adaptation and SCE mechanism

```
Require: NC, S_comp, termination criteria
initialize the position and velocity of all particles
repeat
    E ← sorted X in increasing order according to the functional value
    for each complex $j = 1$ to NC do
        $A_j$ ← divided E into NC complexes
        for each particle $i = 1$ to S_comp in $A_j$ do
            if $f(\vec{X}_i) < f(\vec{P}_i)$ then
                $\vec{P}_i = \vec{X}_i$
                $\vec{w}_i = (\frac{w_{max}+w_{min}}{2} - w_{min}) \cdot \vec{U} + w_{min}$
            else
                $\vec{w}_i = (\frac{w_{max}+w_{min}}{2} - w_{min}) \cdot \vec{U} + (\frac{w_{max}+w_{min}}{2})$
            end if
        end for
        for each particle $i = 1$ to S_comp in $A_j$ do
            $\vec{G} = min\{\vec{P}_0, \vec{P}_1, ..., \vec{P}_{S\_comp}\}$
            $\vec{V}_i = \vec{w}_i \cdot \vec{V}_i + c_1 \cdot \vec{U}_1 \otimes (\vec{P}_i - \vec{X}_i) + c_2 \cdot \vec{U}_2 \otimes (\vec{G} - \vec{X}_i)$
            $\vec{X}_i = \vec{X}_i + \vec{V}_i$
        end for
    end for
until termination criteria is met
```

number of optimization runs was 1 100 (i.e. 11 benchmark functions × 4 PSO variants × 25 program runs).

The second set consists of 120 optimization problems. On 30 datasets of MOPEX catchments, 4 benchmark questions were analysed, which are standard objective functions used for solving inverse problem related to calibrations of hydrological models [9, 29]. Each optimization based on one objective function and one data forcing for one catchment was repeated 25 times. The total number of optimization runs was 12 000 (i.e. 4 objective functions × 30 catchments × 4 PSO variants × 25 program runs).

All algorithms, benchmark functions and Bilan model were written in C++ programming language, and the code ran under 64-bit Linux operating system. All graphs and statistical tests were made in R statistical software environment [133].

4.3.1 The CEC 2005 benchmark optimization problems

In order to compare the optimization algorithms, the single-objective benchmark optimization problems are analysed. In total, 11 benchmark functions from the special session on single optimization problems CEC 2005 [131] were used. This set of benchmark functions were used by many researches for solving optimization problems [18, 67, 80].

Following minimization benchmark problems were tested: Sphere (f_1), Schwefel 1.2 (f_2), Elliptic rotated (f_3), Schwefel 1.2 with noise (f_4), Schwefel 2.6 (f_5), Rosenbrock (f_6), Griewank rotated (f_7), Ackley rotated (f_8), Rastrigin (f_9), Rastrigin rotated (f_{10}) and Weierstrass rotated (f_{11}). The optimization problems f_1 - f_5 are uni-modal functions, the remaining f_6 - f_{11} are multi-modal functions. For formulas of all functions, range of the search space, location of the minimum value and other description, see [131].

The setting of distributed PSO versions follows [40, 43, 67]. The number of complexes is set to 6, and there are 25 particles at each complex. The number of shuffling is 5, and the maximum number of function evaluation is set to $10\,000 \cdot Dim$. The problem space has 30 dimensions due to preservation the setup from [67]. For results analysis, the total number of optimization runs is set to 25, where each run stops when the maximum number of function evaluations is achieved. In terms of the PSO coefficients, the acceleration constants for modifications with inertia weight are $c_1 = c_2 = 2$, and the acceleration constants for adaptation using constriction factor is $c_1 = c_2 = 1.49445$ with constriction coefficient $K = 0.729$.

4.3.2 Optimization of the hydrological model - case study

After comparison of the analysed optimization algorithms on benchmark functions, the developed distributed versions of PSO were applied on solving the real-life optimization related to the estimation of values of parameters of lumped hydrological model Bilan.

The settings of PSO parameters on Bilan optimization problems were selected after running several tests with different parameter settings. The number of complexes is set to 3. Each complex is composed of 40 particles, and the number of generations in one complex is 20. The number of shuffling is 10. To analyse the results, the total number of model runs is 25. The acceleration constants and value for constriction factor are the same as in the single-objective benchmark optimization problems.

4.3.2.1 The hydrological model Bilan

In the case study, the calibration of Bilan rainfall-runoff model is studied. Bilan is a lumped physically-based water balance model developed in T. G. Masaryk Water

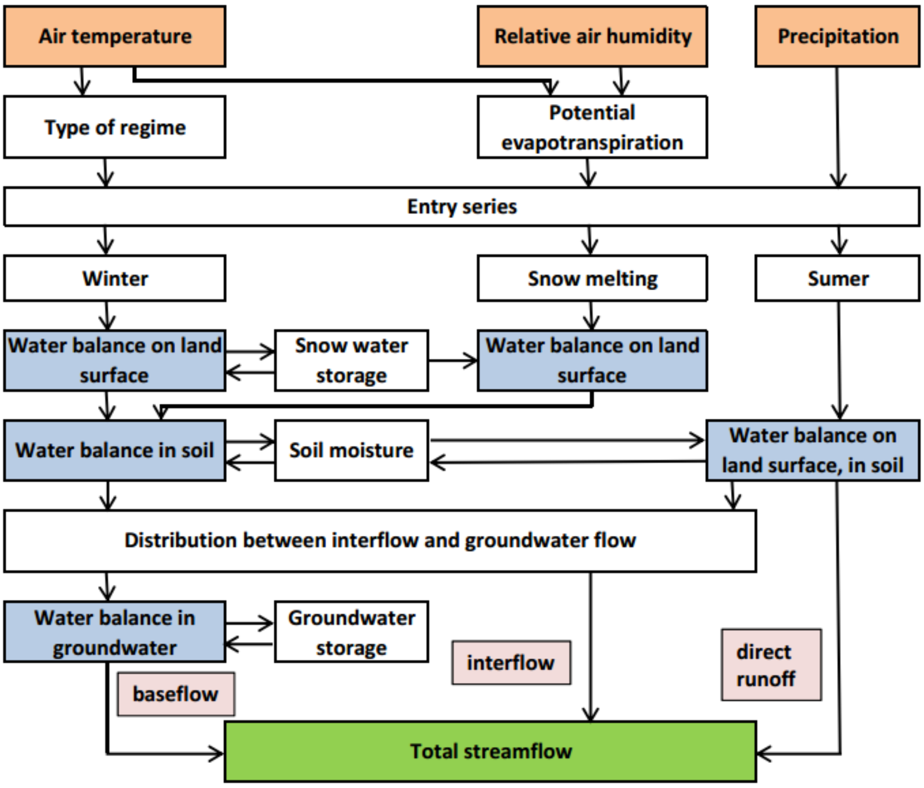

Figure 4.1: Scheme of the Bilan rainfall-runoff model [72]

Research Institute in the Czech Republic [78]. It is a standard tool commonly used for assessment of water balance in the catchment [53, 54, 88].

It is a conceptual hydrological model, which explains the hydrological balance of a catchment using the system of mathematical relationships, which preserve mass balance. It describes basic principles of water balance on ground, in the zone of aeration, including the effect of vegetation cover and groundwater. The main model forcings are time series of precipitation [mm], air temperature [°C] and relative air humidity [%]. Time series of air temperature and relative air humidity are used to estimate the potential evapotranspiration [54, 132].

The temporal dynamics of reservoirs is described by first order differential equations, which are numerically solved using the Euler's method. The calculated total streamflow is given by two components of the river flow. The fast response is simulated through direct runoff reservoir, and the second slow runoff component is explained with baseflow reservoir [61]. The Bilan scheme is shown on Figure 4.1, and further description of the model could be found in [132].

The total amount of parameters of the daily version of the Bilan model is six, and they are listed in Table 4.1. The search space was constrained with physically

Table 4.1: Parameters of the daily version of Bilan model [72]

Name	**Description**	**PC** [1]
Spa	capacity of the soil moisture storage	Spa $\in \langle 0, 200\rangle$
Alf	parameter of rainfal-runoff equation (direct runoff)	Alf $\in \langle 0, 1\rangle$
Dgm	temperature/snow melting factor	Dgm $\in \langle 0, 200\rangle$
Soc	parameter controlling distribution of percolation into interflow and groundwater recharge under summer conditions	Soc $\in \langle 0, 1\rangle$
Mec	parameter controlling distribution of percolation into interflow and groundwater recharge under snow melt conditions	Mec $\in \langle 0, 1\rangle$
Grd	parameter controlling outflow from groundwater storage (baseflow)	Grd $\in \langle 0, 1\rangle$

[1] PC = parameter constraints

meaningful ranges of parameters (see column PC in Table 4.1). The parameter constraints were estimated by an expert knowledge.

4.3.2.2 Objective functions

The Bilan model is calibrated against the observed streamflow data, so the model time series of observed streamflow are used for calibrating of the model. Therefore, different calibration indices based on information obtained from model residuals are used for estimation of Bilan parameters. The solution of related inverse problem, which minimizes the analysed hydrological index - objective function, was used. This approach is a standard way of calibration of lumped hydrological models [2, 15].

The investigated objective functions are in hydrological modelling commonly used accuracy criteria: mean squared error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and Nash-Sutcliffe efficiency (NS) [9, 29, 30].

The analysed objective functions are defined as

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(R[i] - RM[i])^2, \tag{4.12}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|R[i] - RM[i]|, \tag{4.13}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|R[i] - RM[i]|}{R[i]}, \tag{4.14}$$

$$FRV = \frac{\sum_{i=1}^{N} (R[i] - RM[i])^2}{\sum_{i=1}^{N} \left(R[i] - \overline{R}\right)^2}, \tag{4.15}$$

where R is observed streamflow [mm], RM is modelled streamflow [mm], $\overline{R}$ is mean of the observed streamflow [mm] and N is total number of observations.

The fraction residual variance (FRV) was used for the calibration purposes, however, its extension the Nash-Sutcliffe coefficient is used for standardized assessment of the results of calibration of hydrological models [101, 121]. The NS is defined as

$$NS = 1 - FRV. \tag{4.16}$$

4.3.2.3 Dataset

For evaluation of the optimization ability of the proposed PSO modifications, Bilan model was calibrated using datasets from 30 US catchments. The meteorological and hydrological data were obtained from Model Parameter Estimation Experiment project (MOPEX) [35], which serves for benchmarking of hydrological models and calibration approaches.

For the analysis, the daily records from 1948 to 2003 were used. The main meteorological forcings of Bilan model were mean areal precipitation, mean air temperature calculated as an average of given maximum and minimum daily temperature, and potential evaporation. Table 4.2 lists the major characteristics of each catchment including latitude, longitude, drainage area, dominant soil and vegetation types.

4.4 Results

4.4.1 The CEC 2005 benchmark optimization problems

Figure 4.2 presents the convergence graphs for each benchmark function while utilizing all four distributed modifications of the PSO algorithm. The x-axis indicates the number of function evaluations and the y-axis the logarithmic value of the best fitness, i.e. the difference between the searched and the best achieved functional value. A decline with the number of evaluation is clearly visible for LinTimeVarW, AdaptW and APartW modifications in all functions, and it thus indicates the approach to the global optimum. The ConstrFactor variant has the worst performance, where the decline towards the global optimum is not evident.

Table 4.2: Characteristics of the catchments

USGS ID	**Lat.**	**Long.**	**Area [km^2]**	**Soil type**	**Veg. type** [1]
01127000	41.5980	-71.9850	1147	Sandy loam	MF
01197500	42.2320	-73.3550	454	Sandy loam	MF
01321000	43.3528	-74.2708	790	Sandy loam	MF/CS
01371500	41.6860	-74.1660	1144	Silt loam	MF
01372500	41.6531	-73.8731	291	Silt loam	MF
01426500	42.0031	-75.3839	957	Silt loam	MF
01445500	40.8306	-74.9786	171	Sandy loam	MF
01503000	42.0353	-75.8033	3591	Silt loam	MF
01512500	42.2181	-75.8486	2386	Silt loam	MF
01518000	41.9083	-77.1297	454	Silt loam	MF
01531000	42.0022	-76.6350	4032	Silt loam	MF
01534000	41.5583	-75.8950	616	Loam	MF
01541000	40.8969	-78.6772	507	Loam	MF
01541500	40.9717	-78.4061	597	Loam	MF
01543500	41.3172	-78.1033	1102	Loam	MF
01548500	41.5217	-77.4478	972	Silt loam	MF
01556000	40.4631	-78.2000	468	Sandy loam	MF
01558000	40.6125	-78.1408	354	Sandy loam	MF
01559000	40.4850	-78.0190	1313	Sandy loam	MF
01560000	40.0717	-78.4928	277	Silt loam	MF
01562000	40.2158	-78.2656	1216	Silt loam	MF
01567000	40.4783	-77.1294	5397	Sandy loam	MF
01574000	40.0822	-76.7203	821	Silt loam	MF
01610000	39.5369	-78.4578	5002	Loam	MF
01628500	38.3220	-78.7550	1744	Clay	MF/CS
01631000	38.9139	-78.2111	2642	Loam	CS
01634000	38.9767	-78.3364	1236	Loam	CS
01643000	39.3880	-77.3800	1315	Silt loam	MF
01664000	38.5306	-77.8139	998	Clay loam	CS
01668000	38.3222	-77.5181	2568	Clay loam	CS

[1] MF = Mixed forest, CS = Closed shrublands

For the statistical comparison, the non-parametric Wilcoxon test was used according to [33]. Inputs to the testing procedure were the best fitness values achieved for the PSO modifications. The optimization performance of the new proposed APartW was statistically compared only with the AdaptW variant, because the AdaptW was the best modification in research papers of [67, 102], and thus it serves as a benchmark method.

Table 4.3 summarizes results of the Wilcoxon test, and other statistical indices are also indicated. The minimum, 25% quartile, median, 75% quartile, maximum, mean, standard deviation and P value are available. All 25 program runs of each

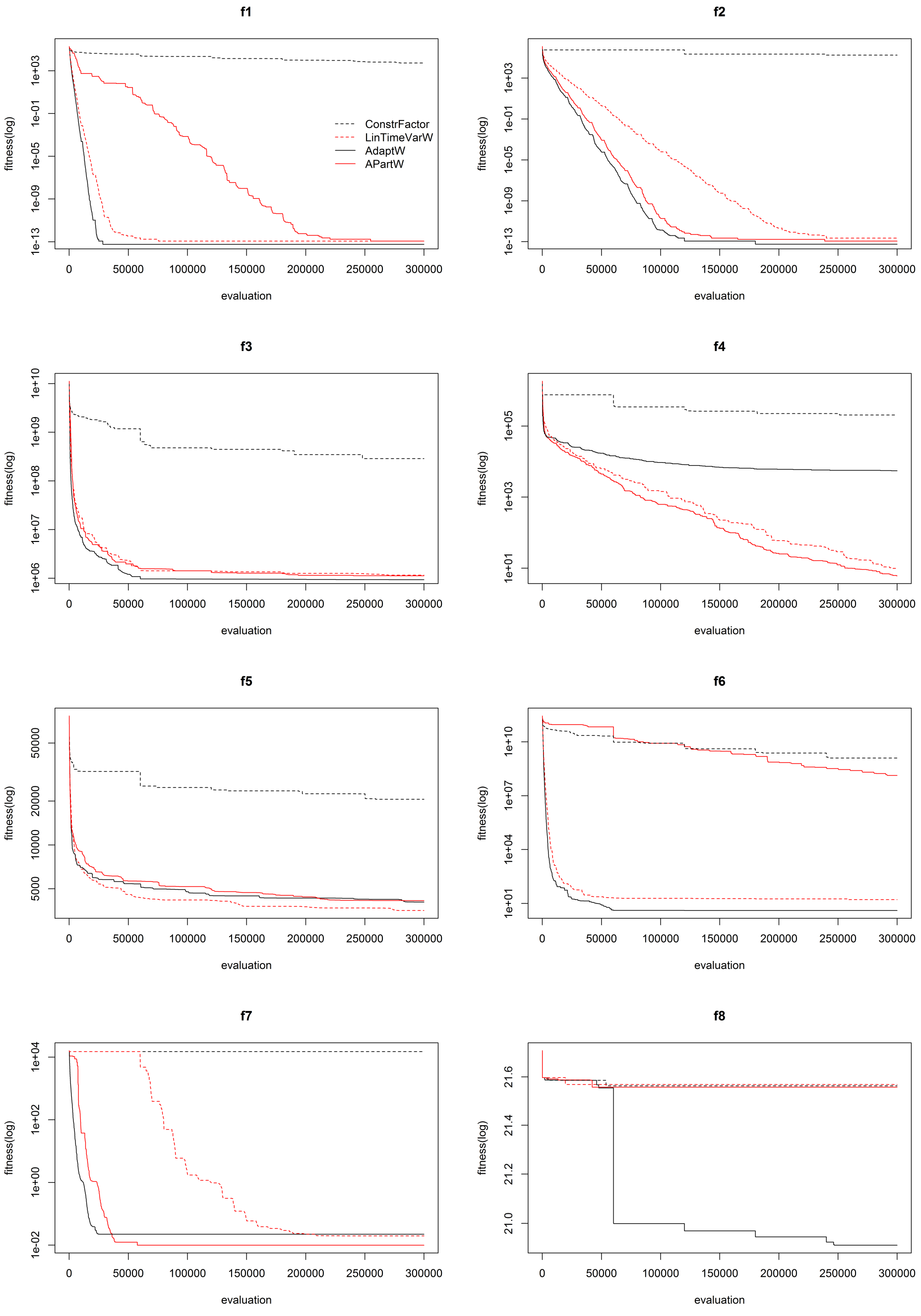

Figure 4.2: Convergence graphs for the best fitness of the ConstrFactor, LinTimeVarW, AdaptW and APartW modifications for evaluated benchmark functions $f_1 - f_{11}$

modification were compared in each numbered evaluation. The values reflect the best achieved fitness, and report other statistical indices belonging to the same numbered evaluation.

The results of the analysis in Table 4.3 show that the APartW modification

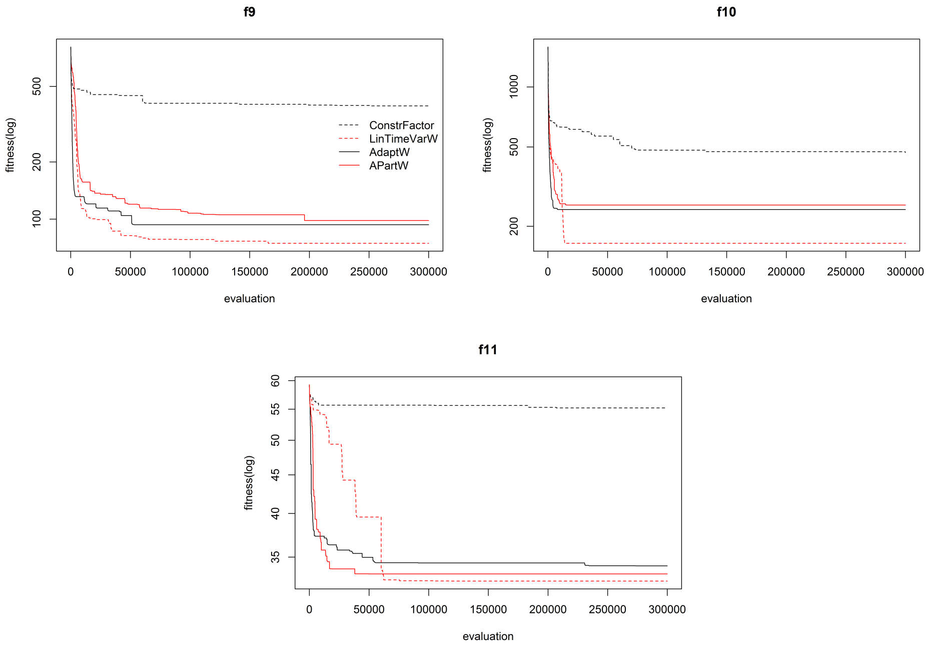

Figure 4.2: Continued

gives significantly better results for three benchmark functions (f_4, f_7 and f_{11}). In functions f_3, f_5 and f_9, there is no significant difference between APartW and AdaptW. Beyond that, in functions f_1 and f_2, both APartW and AdaptW found the global minimum. For multi-modal functions f_6, f_8 and f_{10}, the AdaptW variant gives significantly better results. The differences in the APartW and AdaptW are in agreement with the convergence graphs on Figure 4.2.

Based on the findings, it can be concluded that the APartW version of PSO is comparable with AdaptW modification in optimizing chosen benchmark functions. The APartW is reliable for single-objective optimization, and thus suitable for calibration of the Bilan rainfall-runoff model parameters, where only one objective function at time is minimized.

4.4.2 The calibration of Bilan model

For calibration of the Bilan model, hydrological and meteorological data from 30 US catchments were used. The analysis of the main findings from all catchments is provided, but for clarity, the detailed results from only one catchment are displayed. The chosen catchment (USGS ID 01531000) provides the outline of obtained results, and it serves to perform the overall findings.

The ensemble simulations of the total number of model runs equals to 25 were used. At each ensemble simulation, only one objective function was optimized.

Table 4.3: Statistical indices of 11 benchmark functions for the APartW and AdaptW modifications

Func.	**PSO**	**min**	25%	**median**	75%	**max**	**mean**	**std**	***P* value** [1]
f_1	APartW	5.68E−14	3.98E−13	6.93E−12	8.41E−10	1.94E−05	8.52E−07	3.87E−06	1.000
	AdaptW	0.00E+00	5.68E−14	1.14E−13	1.71E−13	2.27E−13	1.02E−13	6.14E−14	
f_2	APartW	5.68E−14	5.12E−13	9.66E−13	5.46E−12	2.68E−11	3.73E−12	5.77E−12	0.970
	AdaptW	5.68E−14	2.27E−13	3.98E−13	6.82E−13	1.53E−12	5.16E−13	4.33E−13	
f_3	APartW	3.72E+05	1.99E+06	4.05E+06	7.30E+06	2.92E+07	6.33E+06	6.71E+06	**0.439**
	AdaptW	4.17E+05	7.08E+05	1.02E+06	1.57E+06	2.30E+06	1.15E+06	5.78E+05	
f_4	APartW	1.48E−01	3.29E+00	9.75E+00	5.00E+01	1.17E+03	1.02E+02	2.48E+02	**0.000*****
	AdaptW	1.08E+03	4.28E+03	6.66E+03	1.07E+04	1.74E+04	7.62E+03	4.22E+03	
f_5	APartW	2.58E+03	3.32E+03	4.15E+03	4.78E+03	6.10E+03	4.11E+03	9.63E+02	**0.424**
	AdaptW	2.20E+03	3.51E+03	4.07E+03	4.92E+03	6.68E+03	4.26E+03	1.28E+03	
f_6	APartW	3.25E+02	1.01E+05	1.35E+08	1.21E+09	1.03E+10	1.19E+09	2.51E+09	1.000
	AdaptW	1.71E−03	3.83E+00	4.06E+00	7.47E+00	1.80E+01	5.76E+00	4.87E+00	
f_7	APartW	2.84E−14	3.69E−11	1.23E−02	3.45E−02	7.72E+03	3.09E+02	1.54E+03	**0.040***
	AdaptW	2.84E−14	9.86E−03	2.22E−02	3.69E−02	6.15E−02	2.56E−02	1.69E−02	
f_8	APartW	2.08E+01	2.16E+01	2.17E+01	2.18E+01	2.19E+01	2.17E+01	2.03E−01	0.968
	AdaptW	2.06E+01	2.14E+01	2.16E+01	2.18E+01	2.20E+01	2.15E+01	3.89E−01	
f_9	APartW	6.37E+01	8.66E+01	1.08E+02	1.21E+02	1.56E+02	1.05E+02	2.58E+01	**0.770**
	AdaptW	5.67E+01	7.36E+01	9.35E+01	1.17E+02	1.50E+02	9.48E+01	2.57E+01	
f_{10}	APartW	1.60E+02	7.92E+02	1.23E+03	1.41E+03	1.96E+03	1.14E+03	5.13E+02	0.973
	AdaptW	1.34E+02	2.05E+02	2.46E+02	3.03E+02	6.71E+02	2.59E+02	1.05E+02	
f_{11}	APartW	2.73E+01	3.16E+01	3.33E+01	3.58E+01	3.94E+01	3.35E+01	3.30E+00	**0.049***
	AdaptW	2.93E+01	3.44E+01	3.72E+01	3.99E+01	6.30E+01	4.13E+01	1.08E+01	

[1] *P* value of the statistical Wilcoxon test. Non-significant differences in medians are highlighted in bold, significantly better median in terms of the new adaptive PSO APartW is in bold with stars (* for $0.01 < P$ value ≤ 0.05, *** for P value ≤ 0.001)

Table 4.4: Best values of all objective functions achieved in the total set of 30 catchments. The best calculated value for each optimized objective function (OOF) is highlighted in bold

OOF	PSO modif.	MSE	MAE	MAPE	NS
MSE	ConstrFactor	6.674E-01	**3.473E-01**	4.186E-01	7.172E-01
	LinTimeVarW	5.816E-01	3.598E-01	4.041E-01	7.466E-01
	AdaptW	**5.816E-01**	3.594E-01	4.144E-01	**7.466E-01**
	APartW	6.253E-01	3.547E-01	**3.376E-01**	7.297E-01
MAE	ConstrFactor	7.432E-01	3.564E-01	3.380E-01	7.097E-01
	LinTimeVarW	7.776E-01	3.445E-01	**3.380E-01**	**7.339E-01**
	AdaptW	**6.822E-01**	**3.421E-01**	3.420E-01	7.329E-01
	APartW	6.854E-01	3.442E-01	3.484E-01	7.259E-01
MAPE	ConstrFactor	7.800E-01	3.550E-01	3.290E-01	6.808E-01
	LinTimeVarW	8.020E-01	**3.477E-01**	**3.181E-01**	6.962E-01
	AdaptW	8.020E-01	3.480E-01	3.238E-01	6.894E-01
	APartW	**6.849E-01**	3.485E-01	3.300E-01	**7.202E-01**
NS	ConstrFactor	7.268E-01	3.544E-01	**3.380E-01**	7.257E-01
	LinTimeVarW	5.816E-01	3.605E-01	4.015E-01	7.466E-01
	AdaptW	**5.816E-01**	3.594E-01	4.144E-01	**7.466E-01**
	APartW	6.219E-01	**3.504E-01**	3.581E-01	7.333E-01

Within all simulations, the remaining three accuracy criteria were also calculated due to evaluation the overall performance.

The results in Table 4.4 show the best achieved values of objective functions within all 30 catchments. The optimized objective function is in the first column, the calculated criteria are in the last four columns. It is evident that the APartW method provides similar results as the LinTimeVarW version. They both reached the best values in four cases (highlighted in bold in the table). In addition, the ConstrFactor gave the worst results (reaches the best values in two cases), and on the other hand, the AdaptW method achieved the best results (reaches the best values in six cases). Even though, the APartW method doesn't provide the best value of the optimized objective function, it achieves good results for the other criteria which are not optimized.

In order to determine whether the implementation of the distributed PSO modifications into Bilan model increases the model performance, the results were compared with the integrated binary search (BS) optimization technique. The best achieved objective criteria for all catchments using BS method are following: MSE = 1.044E+00, MAE = 3.598E-01, $MAPE$ = 3.287E-01 and NS = 7.462E-01. The results are for optimization based on MSE, MAE, $MAPE$ and NS, respectively. In comparison with Table 4.4, the optimization with PSO method is always better

Table 4.5: The contrast test of the unadjusted medians with ranking. The $Rank$ is ranking based of contrast test, $W.Rank$ is ranking based of Wilcoxon pair test

MSE	ConstrFactor	LinTimeVarW	AdaptW	APartW	Rank	W.Rank
ConstrFactor	-	372.66	373.23	373.22	4	4
LinTimeVarW	-372.66	-	0.57	0.55	3	3
AdaptW	-373.23	-0.57	-	-0.01	1	2
APartW	-373.22	-0.55	0.01	-	2	1
MAE	ConstrFactor	LinTimeVarW	AdaptW	APartW	Rank	W.Rank
ConstrFactor	-	374.78	374.82	374.82	4	4
LinTimeVarW	-374.78	-	0.04	0.04	3	3
AdaptW	-374.82	-0.04	-	0.00	2	2
APartW	-374.82	-0.04	-0.00	-	1	1
MAPE	ConstrFactor	LinTimeVarW	AdaptW	APartW	Rank	W.Rank
ConstrFactor	-	374.96	375.02	375.02	4	4
LinTimeVarW	-374.96	-	0.06	0.05	3	3
AdaptW	-375.02	-0.06	-	-0.00	1	1-2
APartW	-375.02	-0.05	0.00	-	2	1-2
NS	ConstrFactor	LinTimeVarW	AdaptW	APartW	Rank	W.Rank
ConstrFactor	-	375.01	374.91	374.91	4	4
LinTimeVarW	-375.01	-	-0.10	-0.10	1	1
AdaptW	-374.91	0.10	-	-0.00	2	2
APartW	-374.91	0.10	0.00	-	3	3

than with the integrated BS method.

Table 4.5 displays results from the contrast test of the unadjusted medians according to [33]. After pairwise comparison of all PSO modifications, the ranks of each method were determined. The APartW variant achieved the first rank once, the second rank two times and the third rank once. The best method seems to be the AdaptW, which achieved the best results two times and the second rank also two times. On the other hand, the worst is the ConstrFactor version, which was always worse than the others. Additionally, differences in medians between LinTimeVarW, AdaptW and APartW are very small, which indicates similar performances.

In addition to the contrast test, the Wilcoxon pair test of medians according to [33] was conducted. The ranks are displayed in the last column in Table 4.5. The obtained results confirm the results from the contrast test. The differences in the ranks are in the simulations based on MSE and $MAPE$ objective functions, where APartW variant is better than the AdaptW, or as good as AdaptW, respectively. In terms of Wilcoxon test, the APartW is the best modification and ConstrFactor is again the worst.

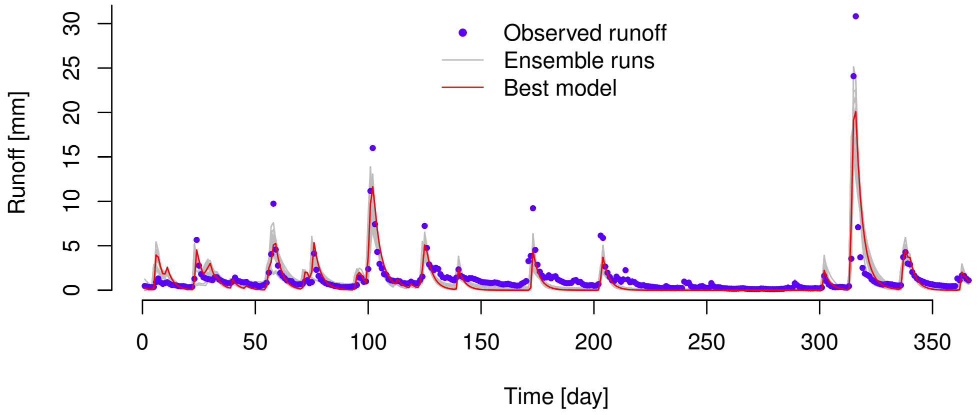

Figure 4.3: Observed streamflow and corresponding simulations from Bilan model using APartW optimization. The optimized objective function was NS. Catchment 01531000, year 1976

Since the APartW modification is a new proposed PSO variant, on Figure 4.3 is displayed the time series of observed and modelled runoff using this method. For clarity, only one chosen year is depicted. The figure gives an example of ensemble simulations with the Bilan model, where the results from the total 25 model runs are coloured in grey. It is evident that the envelope curve of the ensemble simulations would cover most of the observed data points. On the figure, also the streamflow calculated by the best model is plotted (red line), i.e. the simulation with the highest value of NS.

From the Figure 4.3 can be seen that the simulation by the best model underestimates the runoff. This behaviour is also clear from the scatter plot displayed on Figure 4.4, where the regression line lies bellow the $1:1$ line. The corresponding residuals, i.e. differences between observed and simulated streamflow, are depicted on Figure 4.5. The range of the residuals is approximately $[-13.7, 18.3]$ for ensemble runs and $[-7.4, 10.7]$ for the best model.

The values of parameters obtained by the best model of each PSO modification for the chosen catchment are in Table 4.6. The obtained Nash-Sutcliffe efficiency values are also indicated in the the table. It is seen that the NS are very similar for all PSO modifications. The APartW variant achieved the smallest NS value, but it was not significantly smaller than the others.

The results shown that the PSO versions using adaptive inertia weight for calibration of the Bilan model give better results than other PSO modifications.

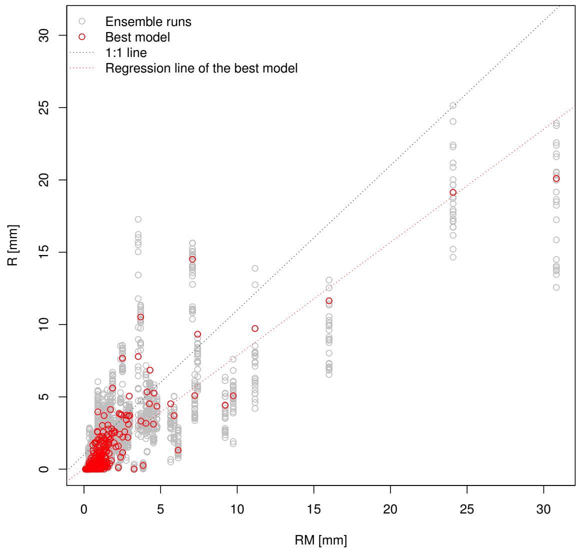

Figure 4.4: Scatter plot of observed and modelled runoff using APartW optimization. The optimized objective function was NS. Catchment 01531000, year 1976

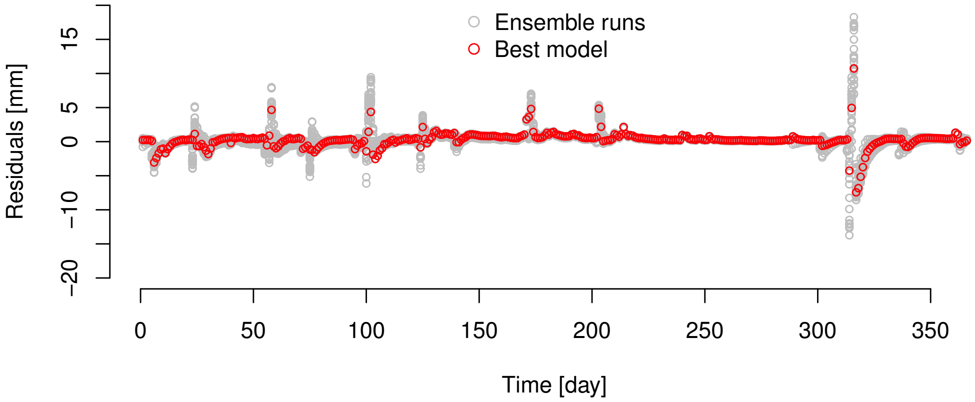

Figure 4.5: Calculated residuals using APartW optimization. The optimized objective function was NS. Catchment 01531000, year 1976

Table 4.6: Parameters of Bilan model for catchment 01531000 obtained by the best model of each PSO modification. The optimized objective function was NS, and the last column indicates the value of that criterion

PSO modif.	Spa	Alf	Dgm	Soc	Mec	Grd	NS
ConstrFactor	198.7	0.091	1.55	7.05E-3	10.68E-3	0.352	0.622
LinTimeVarW	199.9	0.559	1.59	2.33E-3	6.00E-3	0.276	0.644
AdaptW	200.0	0.552	1.59	2.38E-3	6.09E-3	0.275	0.644
APartW	178.3	0.186	1.33	5.85E-3	17.33E-3	0.301	0.621

It is in agreement with Nickabadi et al. [102]. It was found out that the linearly decreasing inertia weight for updating the particle's velocity is more promising than the constriction factor, which is in contradiction with Eberhart and Shi [40], but in this chapter, the distributed versions were used.

4.5 Discussion

4.5.1 The CEC 2005 benchmark optimization problems

Both AdaptW and APartW modifications obtained the global minimum, where the error value was less than 10^{-8}, in three functions. These functions are f_1, f_2 (uni-modal), and f_7 (multi-modal), and the results are comparable with [67].

Better results were obtained by Liang and Suganthan [80], who applied distributed PSO algorithm with local search. They achieved the global optimum in functions f_1, f_2, f_3, f_6 and f_7. The swarm was dynamic with frequent regrouping and the swarm's size was very small.

Bui et al. [18] achieved the global optimum in the uni-modal functions f_1, f_2 and f_4 using different versions of PSO. Their algorithms did not converge to the global minimum in any multi-modal function, and thus, distributed PSO versions from this chapter can be considered as better methods.

The results are similar to results obtained by Nickabadi et al. [102]. Their algorithms achieved the global optimum in functions f_1, f_2 and f_8.

When comparing the results with another optimization techniques within the CEC 2005 benchmark problems, the distributed PSO versions give comparable results as CoEVO algorithm [118]. They also obtained the global optimum in functions f_1, f_2 and f_7. Results obtained in this chapter are better than the DE algorithm of Rönkkönen et al. [122], which achieved the global optimum in functions f_1 and f_7, or better than BLX_MA method of Molina et al. [97], which converged only in functions f_1 and f_9.

4.5.2 The calibration of Bilan model

The best obtained NS values (> 0.74, see Table 4.4) are comparable to a work of Brauer et al. [16, 17], who implemented the hydroPSO package [154] for calibration of the Wageningen lowland runoff simulator (WALRUS). The WALRUS is a rainfall-runoff model often used in sloping lowland catchments. The differences in their work is that during the calibration of the model, they used 1 year of discharge observations and optimized 4 parameters. Their best achieved NS values for two catchments were 0.87 and 0.83. In this chapter, 30 years of observation were used, and 6 parameters for 30 catchments were calibrated.

Similar values of NS criteria were also obtained by Jiang et al. [69] in calibration of the Xin'anjiang hydrological model, by Lawrence et al. [79] in calibration of the HBV model, or by Zhang et al. [157] in calibration of SWAT hydrological model. Thus, the optimized parameters of the Bilan model can be considered as sufficient estimation, and therefore the used PSO distributed versions are suitable for the calibration.

The Bilan model was extended by a shuffled complex differential evolution (SCDE) global optimization method and the model was applied on 234 catchments in the Czech Republic. The best achieved NS values during calibration of the Bilan model using SCDE was between 0.78 and 0.80, whereas using existing integrated local optimization technique with expert knowledge was 0.73 [87]. Results obtained in this chapter were comparable with this research, however, different dataset was used.

To determine if the distributed versions of PSO improve the model performance, the results were compared with the binary search method. The binary search is a default optimization technique integrated in the Bilan model. It was found out that the objective criteria obtained from the PSO optimization gave better results. Therefore, the implementation of the PSO into the Bilan model improves the model simulations.

4.6 Conclusions

The main aim of this chapter was to test 4 selected PSO distributed versions on single-objective benchmark optimization problems, and to apply them on calibration of hydrological model Bilan. For all 4 PSO versions, 3 275 optimization problems were analysed, in which 275 minimizations for benchmark problems (i.e. 11 benchmark function × 25 program runs) and 3 000 inverse hydrological problems (i.e. 4 objective functions × 30 catchments × 25 program runs) were solved.

The new proposed variant APartW was compared with other existing PSO modifications - ConstrFactor, LinTimeVarW and AdaptW on 11 benchmark functions prepared for the special session on real-parameter optimization of CEC 2005. The APartW version is comparable with the AdaptW and LinTimeVarW variants, whereas the ConstrFactor had the worst performance.

The statistical comparison of AdaptW and APartW modifications shown that both methods obtained the global minimum in three functions (f_1, f_2 and f_7). The APartW variant gave significantly better results in functions f_4, f_7 and f_{11}. Beyond that, in functions f_3, f_5 and f_9, there was no significant difference between the APartW and AdaptW method.

All four PSO modifications were implemented into the Bilan rainfall-runoff model for solving inverse hydrological problems. Based on the contrast test of the unadjusted medians and Wilcoxon test, it was found out that the APartW and AdaptW variants provided the best results. The ConstrFactor performed the worst.

The results highlighted that distributed versions of PSO are promising in single-objective optimization problems. It was confirmed that adaptive variants of the inertia weight are better then linearly decreasing weight. It was also found out that the PSO modifications with parameters of inertia weight give significantly better results than the variant with constriction factor.

The results of this chapter extended the range of utilization of the PSO global optimization techniques. The performance of the distributed versions of PSO is promising, and the algorithms can be implemented into other real optimization problems.

CHAPTER 5

Combination of hybrid artificial neural networks with particle swarm optimization algorithm for SPEI forecasting

Abstract

Prediction of drought is very important for prevention of the potential threat of the drought event. The recent climatic water balance indicator, the Standardized precipitation evapotranspiration index (SPEI), was forecasted within this chapter. A new tool for the SPEI simulations was proposed, which is a combination of hybrid artificial neural networks (ANN) with particle swarm optimization (PSO). The PSO algorithm was used for training of the model weights to achieve higher accuracy in shorter computational time. In this research, the influence of chosen PSO modifications, number of inputs into the ANN, number of neurons in the hidden layer, and influence of the type of optimized objective function on modelled SPEI drought index were evaluated. The case study was conducted on selected set of 8 US catchments with the data of meteorological observations obtained from MOPEX database. It was found out that the best optimization technique is *APartPSO*, and the Nash-Sutcliffe efficiency as an optimized objective function is

This chapter is based on the manuscript: Jakubcová M., Máca P., and Pech P., 2015: Combination of hybrid artificial neural networks with particle swarm optimization algorithm for SPEI forecasting. *Applied Soft Computing*.

the most effective. The results show that the combination of ANN with PSO is suitable for forecasting the SPEI drought index, and thus, tested PSO versions can be used for solving related inverse problems.

5.1 Introduction

There is no universal definition of drought. In general, drought is a climatic event, which results in deficit of water supply, and impacts many areas. The most severe damages can be observed in the field of agriculture, water management, economics, ecosystems, and wildlife [99, 115, 120].

There exist several types of droughts. The meteorological drought is usually defined as period with below-average precipitation, and general lack of moisture in the atmosphere. In agricultural drought, the deficit of precipitation influences the soil moisture, and thus, mainly crops are affected. The hydrological drought is associated with deficiency of surface water and groundwater storages. The meteorological and hydrological conditions can affect the drinking water, hydroelectric power, or food supply, and then it is defined as socioeconomic drought [145].

Drought severity is directly related to the impacts of drought [58]. It can be described through drought indices. They are usually represented in the form of time series, and in many cases, they are based on actual measured meteorological data. One of the first developed comprehensive drought index was Palmer drought severity index. It is used for analyses of the drought based on water balance equation, and it classifies the weather conditions from extremely wet to extreme drought [106]. Less complex drought indicator is Standardized precipitation index (SPI), which is based on probability of precipitation [92].

More recent variant of the SPI indicator is Standardized precipitation evapotranspiration index (SPEI). Many recent studies [56, 60, 98] devoted into developing new computational tools for simulating the SPEI index, and thus predict the potential threat of the drought. Therefore, the SPEI was forecasted within this chapter using combination of artificial neural networks and particle swarm optimization.

Drought can be forecasted through solving inverse problems using data-driven black box model. One of the suitable tool for simulations are artificial neural networks (ANN). They were applied in many case studies. For example Belayneh and Adamowski [8] forecasted SPI through three different data-driven models on case study in Ethiopia. Their results show that the coupled wavelet neural network models were the most effective for SPI predictions.

Deo and Şahin [32] used ANN models for forecasting SPEI index for catchments in Australia, and they confirmed that artificial neural networks are a useful tool for prediction of the chosen drought index. In the study of Hosseini-Moghari and Araghinejad [63], the ANN models were applied for drought forecasting in Iran, and suitable performance was achieved using statistical neural networks.

Training weights of the model in neural networks is a matter of solving inverse problems. The weights are standardly trained by back-propagation learning [123]. In recent studies, the back-propagation technique was compared with particle swarm optimization (PSO) [20, 44, 74].

Chau [21] proposed a model with multilayer perceptron trained by PSO, and he applied the model in the catchment of the Shing Mun River for estimation of water level. The results of his study show that the model achieved higher accuracy in shorter computational time compared to the method of back-propagation. Good results were also obtained by Senthil Babu and Vinayagam [124], who combined PSO with ANN for prediction of the roughness of the surface. The forecast of precipitation with ANN trained by PSO was also sufficient and accurate [59, 147].

The performance of the ANN model can be improved by integrating more neural networks into one hybrid ANN. The integration can be in terms, that outputs from several ANN are used as inputs into one ANN model. The integrated ANN were used in different hydrological case studies. For example Nourani and Kalantari [104] applied them for simulating the rainfall-runoff sediment processes, or Huo et al. [64] and Wang et al. [141] applied this technique for estimating the river streamflow.

The main aim of this chapter is to combine hybrid artificial neural networks with particle swarm optimization. The PSO technique serves for training of the model weights. The integrated ANN models are developed and applied for forecasting the SPEI drought index. The case study is conducted on 8 US catchments with 54 years of observations.

The rest of this chapter is organized as follows. Section 5.2 explains the methodology with respect to the analysed dataset, SPEI drought index, and settings of the ANN models. Section 5.3 summarizes the results, which are discussed in Section 5.4. The main findings are concluded in Section 5.5.

5.2 Methodology

The combination of hybrid artificial neural network models with particle swarm optimization technique was applied for forecasting the SPEI drought index. The

Table 5.1: Characteristics of selected catchments

USGS ID	Lat.	Long.	Area (km^2)	mP (mm)	mET (mm)
01127000	41.5980	-71.9850	1 147	99.88	60.20
01197500	42.2320	-73.3550	454	99.92	55.03
01321000	43.3528	-74.2708	790	103.33	57.49
01371500	41.6860	-74.1660	1 144	94.71	63.10
01372500	41.6531	-73.8731	291	86.83	58.40
01426500	42.0031	-75.3839	957	91.28	59.01
01445500	40.8306	-74.9786	171	104.30	65.72
01503000	42.0353	-75.8033	3 591	85.91	59.24

models differ in four variables - in number of inputs, number of neurons in hidden layer, PSO method used for training, and optimized objective function.

5.2.1 Dataset

For ANN model simulations, datasets from 8 US catchments were used. The meteorological data were obtained from Model Parameter Estimation Experiment project (MOPEX), which serves for benchmarking of hydrological models [35].

Table 5.1 lists the major characteristics of each of the 8 catchments including latitude, longitude, drainage area, month average precipitation (mP) and month average evapotranspiration (mET) during the observed period.

Dataset from 1948 to 2002 was analysed. The calibration period consists of the data from 1948 to 1975, whereas the validation period from 1976 to 2002. The daily records of precipitation and evapotranspiration from the MOPEX database was modified into monthly time step, from which the SPEI drought index was calculated.

A non-linear transformation on the original data D_o was applied before they were used in ANN model. The transformed data D_t are calculated as

$$D_t = 1 - exp(-0.15 \cdot D_o). \tag{5.1}$$

All values of D_t are thus normalized for the input to the ANN model. Before calculating the accuracy criteria during training of the ANN, the output from the model has to be transformed back into the original data as

$$D_o = \begin{cases} \frac{1}{0.15} \cdot ln\left(\frac{1}{1-D_t}\right) & \text{if} \quad (1-D_t) \neq 0 \\ \frac{1}{0.15} \cdot ln(10E+08) & \text{if} \quad (1-D_t) = 0 \end{cases} . \tag{5.2}$$

5.2.2 SPEI drought index

The artificial neural networks were applied for forecasting the Standardized precipitation evapotranspiration drought index. The input data for calculating SPEI index are monthly time series of climatic water balance, where the potential evapotranspiration is subtracted from precipitation [136].

The SPEI is a variant of the Standardized precipitation drought index, but it takes the influence of potential evapotranspiration into account [136]. It is based on the cumulative probability of a given climatic water balance event occurring at the location. The probabilities are then converted into standard normal distribution, which creates the final value of the SPEI criterion [92, 130].

In this study, the monthly temporal resolution of the data was used with the time scale of the accumulation period equal to 12 months, and type of kernel unshifted rectangular. The applied probability distribution was log-Logistic, and the parameters were fitting based on unbiased probability weighted moments according to [7]. For estimation of the SPEI index, the package *SPEI* [6] in the R programming language was used.

5.2.3 ANN models

The architecture of the applied artificial neural network models is a multilayer perceptron (MLP), which is a suitable universal approximator [62]. The MLP with one input layer, one hidden layer of neurons, and one output layer with one output neuron has been applied in this study. The topology is fully connected, and transfer of information is feedforward.

In this chapter, processing elements (i.e. components of ANN where the computations are performed) with external biases [43] were used. The normalized transformed simulated output MOD_t from the ANN model at given time interval is calculated as

$$MOD_t = w_b + \sum_{j=1}^{N_{hd}} w_j \cdot f(a), \tag{5.3}$$

where w_b is weight of the bias neuron entering the output neuron, w_j is weight of j^{th} neuron in the hidden layer entering the output neuron, $f()$ is activation function, a is activation, and N_{hd} is number of neurons in the hidden layer.

The activation function of neurons is the RootSig, which was chosen according to [37, 86]. The activation function is in general

$$f(a) = \frac{a}{1+\sqrt{1+a^2}}, \tag{5.4}$$

with the activation defined as

$$a = w_{bj} + \sum_{i=1}^{N_{in}} w_{ij} \cdot OBS_{ti}, \tag{5.5}$$

where w_{bj} is weight of the bias neuron entering the j^{th} neuron in the hidden layer, w_{ij} is weight of the i^{th} input entering the j^{th} neuron in the hidden layer, OBS_{ti} is the i^{th} transformed input into the network, and N_{in} is number of inputs in the ANN model.

The weights in ANN models were trained with the PSO optimization technique. As optimized objective criteria serve 5 different statistics, which are often used in hydrological modelling. For SPEI simulations, the integrated neural network models with different settings were used. The PSO methods, objective functions, and integrated ANN models are explained in the next part of this chapter.

All algorithms of the ANN models trained by PSO technique were written in C++ programming language, and the code ran under 64-bit Linux operating system. All post-processing calculations were made in R statistical software environment [133].

5.2.3.1 PSO variants for ANN training

Particle swarm optimization technique was used for training the artificial neural network models. The optimization process is based on iterative procedure of population of particles. Each particle in the population flies through the parametric space, and tries to find the optimal solution [73].

The velocity of particles is updated for all $i = 1, ..., S$, where S is total number of particles in the swarm population. The velocity equation is

$$\vec{V}_i^{t+1} = W \cdot \vec{V}_i^t + c_1 \cdot \vec{U}_1^t \otimes \left(\vec{P}_i^t - \vec{X}_i^t\right) + c_2 \cdot \vec{U}_2^t \otimes \left(\vec{G}^t - \vec{X}_i^t\right), \tag{5.6}$$

where W is parameter of inertia weight, c_1 and c_2 denotes acceleration constants predefined by the user, $\vec{U}_1$ and $\vec{U}_2$ are independent random vectors sampled from a uniform distribution in the range $[0,1]$, $\vec{P}_i = (p_{i1}, p_{i2}, ..., p_{iDim})$ is the best particle's position achieved so far representing the cognition knowledge of swarm particles, and $\vec{G} = (g_1, g_2, ..., g_{Dim})$ is the best location of all particles achieved so far controlling the social influence of swarm particles [73, 128].

Particle's position in the space is calculated as

$$\vec{X}_i^{t+1} = \vec{X}_i^t + \vec{V}_i^{t+1}. \tag{5.7}$$

Table 5.2: Applied PSO modifications for training ANN models

PSO	Inertia weight
LinPSO	$\overrightarrow{W}_i^t = \frac{t_{max}-t}{t_{max}} \cdot (W_{max} - W_{min}) + W_{min}$
ChaoPSO	$\overrightarrow{W}_i^t = 0.5 \cdot \overrightarrow{U}^t + 0.5 \cdot z$
NonlinPSO	$\overrightarrow{W}_i^t = W_{ini} \cdot u^t$
AdaptPSO	$\overrightarrow{W}_i^t = (W_{max} - W_{min}) \cdot \overrightarrow{Ps}^t + W_{min}$
APartPSO	$\overrightarrow{W}_i^t = \left(\frac{W_{max}+W_{min}}{2} - W_{min}\right) \cdot \overrightarrow{U}^t + W_{min}$ $\overrightarrow{W}_i^t = \left(\frac{W_{max}+W_{min}}{2} - W_{min}\right) \cdot \overrightarrow{U}^t + \left(\frac{W_{max}+W_{min}}{2}\right)$

Each position in the population represents a vector of searched parameters. With respect to neural network model, the parameters are the weights of the ANN model.

In this chapter, 5 PSO modifications were used for training the ANN models. All PSO versions update the particle's velocity via parameter of inertia weight, and all of them were compared within the previous research in [67] or [68]. All used PSO methods with the corresponding formulas for calculating the inertia weight are listed in Table 5.2.

The PSO method with linearly decreasing inertia weight (*LinPSO*) was first used by [127]. The parameter of W decreases linearly during iterations from $W_{max} = 0.9$ to $W_{min} = 0.4$. The random chaotic model (*ChaoPSO*) was used by [45]. The inertia weight is changing during iterations based on a random number $\overrightarrow{U}^t$ from the range $[0,1]$, and auxiliary variable $z = 4 \cdot z \cdot (1-z)$, where the initial value of z is uniformly distributed in $[0,1]$. The PSO method with non-linearly decreasing inertia weight (*NonlinPSO*) uses parameter u, which is set to 1.0002, and initial value of inertia weight W_{ini} uniformly distributed in $[0,1]$ [102].

Moreover, two adaptive strategies of inertia weight were used. First, the *AdaptPSO* proposed by [102] calculates the inertia weight through success percentage of the swarm ($\overrightarrow{Ps}^t$). It is calculated as

$$\overrightarrow{Ps}^t = \frac{\sum\limits_{i=1}^{n} \overrightarrow{S}_i^t}{n}, \tag{5.8}$$

where n is size of the population, and $\overrightarrow{S_i}$ shows the success of i^{th} particle, which is equal to 1 if $f(\overrightarrow{P}_i^t) < f(\overrightarrow{P}_i^{t-1})$, or equal to 0 if $f(\overrightarrow{P}_i^t) = f(\overrightarrow{P}_i^{t-1})$. In this approach, $W_{min} = 0$ and $W_{max} = 1$.

Second adaptive strategy is *APartPSO* modification proposed by [68]. It combines the global exploration and local exploitation in the space according to the development of particle's location. If the particle improves its position compared to its previous locations, the first equation from Table 5.2 is used, otherwise, the second equation is applied. In this variant, $W_{min} = 0.1$ and $W_{max} = 0.9$.

The settings of the PSO optimization is following. The distributed versions of PSO were used [67, 68, 152], where the number of complexes is set to 3. At each complex is 40 particles, and the number of generations in one complex is 20. The number of shuffling and redistribution of particles in complexes is 10. The total number of optimization runs is set to 25, where each run stops when the maximum number of function evaluations is achieved. In terms of the PSO coefficients, the acceleration constants are $c_1 = c_2 = 2$ according to [40].

5.2.3.2 Objective functions

The optimized objective functions (OOF) are in hydrological modelling commonly used accuracy criteria [9, 29]. The evaluated criteria are mean squared error (MSE), Nash-Sutcliffe efficiency (NS), persistence index (PI), and two combined accuracy indexes ($cAI1$, $cAI2$). The $cAI1$ and $cAI2$ are composed of two different objective functions, which both influence the final criterion with a given weight.

The formulas of analysed objective functions are defined bellow, where OBS is observed SPEI, MOD is modelled SPEI, $\overline{OBS}$ is mean of the observed SPEI, and n is total number of observations. Consider

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(OBS[i] - MOD[i])^2, \tag{5.9}$$

$$NS = 1 - \frac{\sum_{i=1}^{n}(OBS[i] - MOD[i])^2}{\sum_{i=1}^{n}\left(OBS[i] - \overline{OBS}\right)^2}, \tag{5.10}$$

$$PI = 1 - tPI, \tag{5.11}$$

$$cAI1 = 0.85 \cdot tPI + 0.15 \cdot MSE, \tag{5.12}$$

$$cAI2 = 0.85 \cdot MAE + 0.15 \cdot dRMSE, \tag{5.13}$$

where tPI is transformed persistence index, MAE is mean absolute error, and $dRMSE$ is root mean squared error in derivatives. The equations are

$$tPI = \frac{\sum_{i=1}^{n}(OBS[i] - MOD[i])^2}{\sum_{i=1}^{n}(OBS[i] - OBS[i-1])^2}, \tag{5.14}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|OBS[i]-MOD[i]|, \tag{5.15}$$

$$dRMSE = \sqrt[4]{\frac{1}{n-1}\sum_{i=1}^{n-1}((OBS[i]-OBS[i+1])-(MOD[i]-MOD[i+1]))^4}. \tag{5.16}$$

All above mentioned OOF were used for ANN training during calibration of the model, but also for results analyses. The objective functions serve as accuracy criteria for evaluation of the models performance.

5.2.3.3 Integrated ANN models

In this study, always 5 artificial neural network models were integrated into one hybrid ANN model (hANN). The outputs from four models are inputs into the fifth model as it is displayed on Figure 5.1. The final forecasted SPEI drought index is the output from the fifth ANN model.

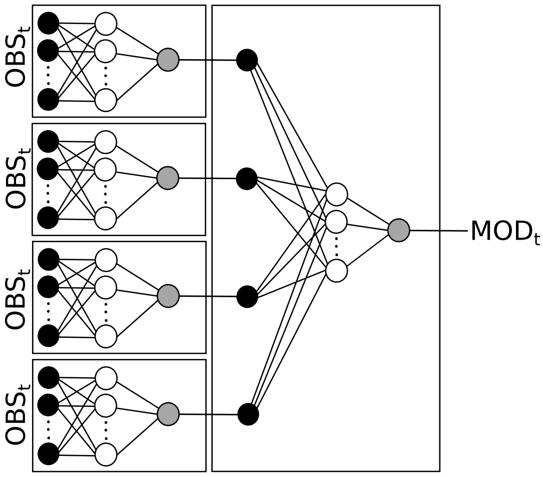

Figure 5.1: Integrated ANN models into hANN. Circles filled with black represent input layer, circles filled with white represent hidden layer, and circles filled with grey represent outputs

The hANN models differ in number of inputs, number of neurons in hidden layer, PSO method used for training, and optimized objective function. The N_{in} is equal to 6 or 12 with the input variables according to Table 5.3, where SPI is Standardized precipitation drought index obtained from precipitation data. The N_{hd} is either 3 or 6. The PSO method used for training is the same for the whole hANN, and it is one of the five variants mentioned above in the text.

The optimized objective criteria is different for each of the ANN model within the hANN. It is the main advantage of the hANN, because the performance of

Table 5.3: Input variables into the ANN models for each number of inputs

N_{in}	**Input variable**
6	SPEI[$t-c$], for $c = 1,2,...,6$
12	SPEI[$t-c$], for $c = 1,2,...,12$
12	SPI[$t-c$], for $c = 1,2,...,12$

each model is combined, and the fifth ANN model works as a correction model [52, 64].

The general form of the hANN model within this chapter is N_{in}-N_{hd}-PSO-OOF, e.g. 6-3-$LinPSO$-MSE. The OOF in the notation is the objective function optimized by the final fifth ANN model. To distinguish between input variable SPEI and SPI for $N_{in} = 12$, $N_{in} = 12$ was used for SPEI, and $N_{in} = 12s$ for SPI in the notation of the hANN. The PSO and OOF in the notation in validation period indicate the optimization method and objective function used for obtaining model weights during calibration.

The total number of hANN models for each catchment is 150 (i.e. 3 sets of inputs × 2 sets of neurons in hidden layer × 5 PSO variants × 5 optimized objective functions).

5.3 Results

To obtain a representative set of results for the analysis, each model runs 25 times for calibration and once for validation. The total number of runs was 31 200 (i.e. 3 sets of inputs × 2 sets of neurons in hidden layer × 5 PSO variants × 5 objective functions × 8 catchments × 26 model runs).

5.3.1 Statistical evaluation

The analysed catchment, number of inputs, number of neurons in hidden layer, optimized objective function, and PSO variant were considered as factors influencing the resulted accuracy criteria. Obtained data were analysed based on statistical tests, and the results are summarized in tables and illustrated on figures.

ANOVA technique was used in order to statistically evaluate obtained results. Table 5.4 shows the significance level for each factor based on analysis of variance. If there is a significant difference with a level of $\alpha = 0.05$ (i.e. at least one star in the table), it means that at least one level of the given factor gives signifi-

Table 5.4: Influence of each factor on the accuracy criteria. Results for calibration are on the left, for validation on the right side of each column. The significance level *** is for P value ≤ 0.001, ** for 0.001 < P value ≤ 0.01, * for 0.01 < P value ≤ 0.05, · for 0.05 < P value ≤ 0.1, and no sign for P value > 0.1

Factor	Accuracy criteria									
	MSE		NS		PI		cAI1		cAI2	
	cal.	val.	cal.	val.	cal.	val.	cal.	val.	cal.	val.
Catch.	***	***	***	***	***	***	***	***	***	***
N_{in}	***	*		***	***	***	***	·	***	***
N_{hd}	***	***	***	***	***	***	***	***	***	
OOF	***	***			***	·	***	***	***	
PSO	·		***	**	***	***	*			

cantly different results. For example, all accuracy criteria depend on the choice of catchment, whereas the lower significance levels were achieved for PSO selection.

Table 5.4 is completed with Figures 5.2 and 5.3, where box plots of chosen accuracy criteria are displayed. The MSE and NS criteria were selected for representation, since they are the most commonly used accuracy statistics in hydrological modelling.

From the figures of box plots can be estimated, which level of each factor gives significantly different results than the others. For both calibration and validation seems to be a better choice N_{hd} = 6, and $APartPSO$ modification, whereas the OOF = MSE seems to be worse than the others. Choice of N_{in} depend on the calibration (N_{in} = 12 or 12s) and validation (N_{in} = 6), as is the choice of catchment.

Statistics were applied to support the estimation of the best level of each factor, and to generalize the conclusion for all accuracy criteria. The applied statistical technique was post hoc Tukey's HSD (honest significant difference) test for multiple comparison of means [105].

The best levels of each factor obtained during calibration and validation reflects Table 5.5. It is evident, that some levels are significantly better for simulations, but sometimes there is no difference between two or more levels. Based on the results, the best hANN models were determined.

For calibration, there are two hANN models with two different N_{in} with the same simulation ability. The superior are 12 inputs into the neural networks with 6 neurons in the hidden layer optimized by NS criteria with $APartPSO$ method.

For validation, there exist six hANN models with three OOF and two PSO factors, whose performances are not different. The best results were obtained by models with 6 SPEI inputs and 6 neurons in the hidden layer.

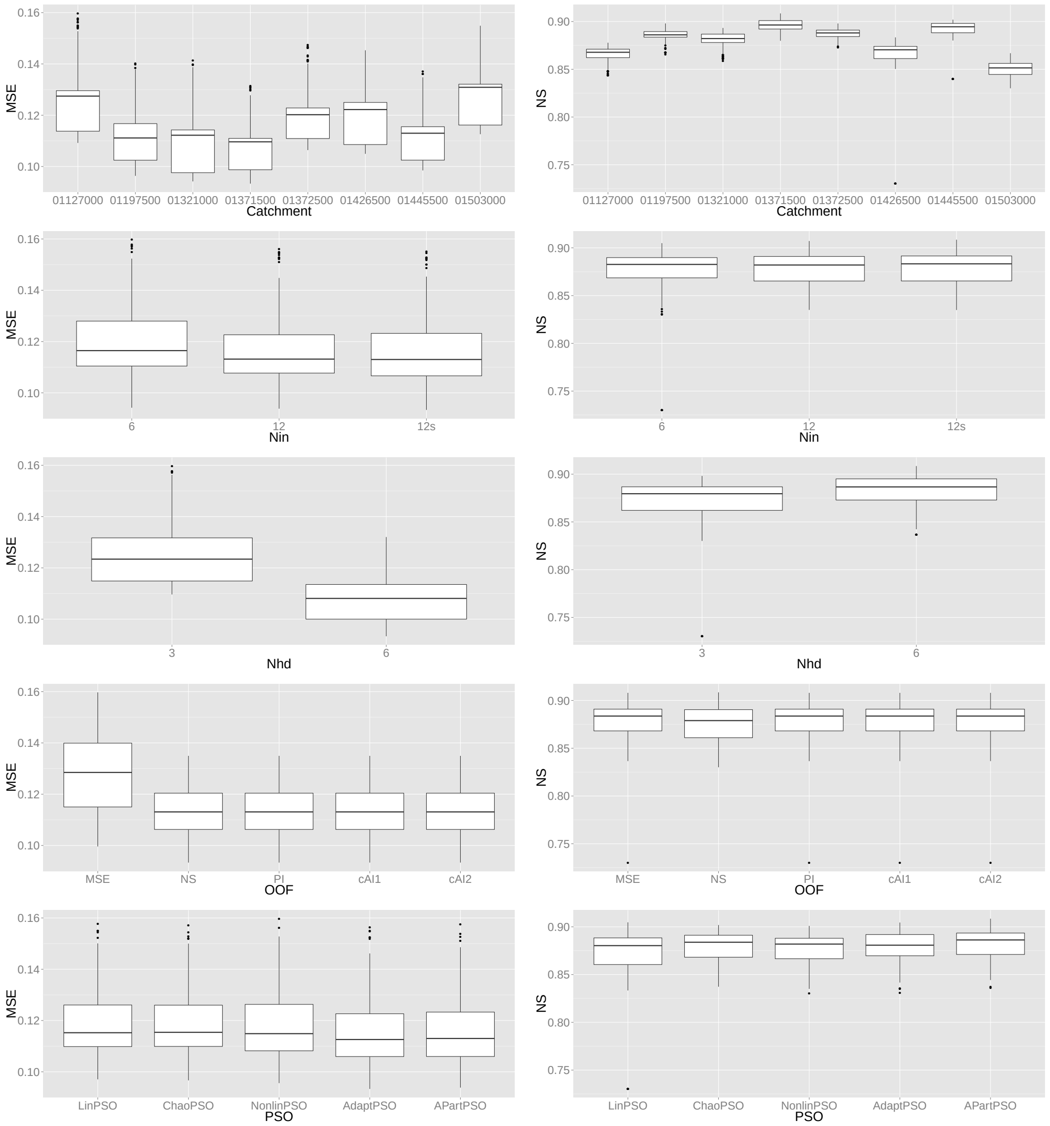

Figure 5.2: Box plots of all achieved values of MSE (left column) and NS (right column) during calibration based on different factors

5.3.2 Best hANN models

From the statistical evaluation, 2 overall best hANN for calibration, and 6 overall best hANN for validation out of the total 150 possible hANN models were estimated. For clarity, one best hANN model for calibration and one for validation were selected. The models are 12-6-$APartPSO$-NS and 6-6-$AdaptPSO$-NS, respectively. These two models are representative for the given time period.

The overall performance of the hANN is explained by statistical indices of the model accuracy criteria obtained during calibration and validation. Table 5.6 displays the minimum, 25% quartile, median, 75% quartile, maximum, mean, and

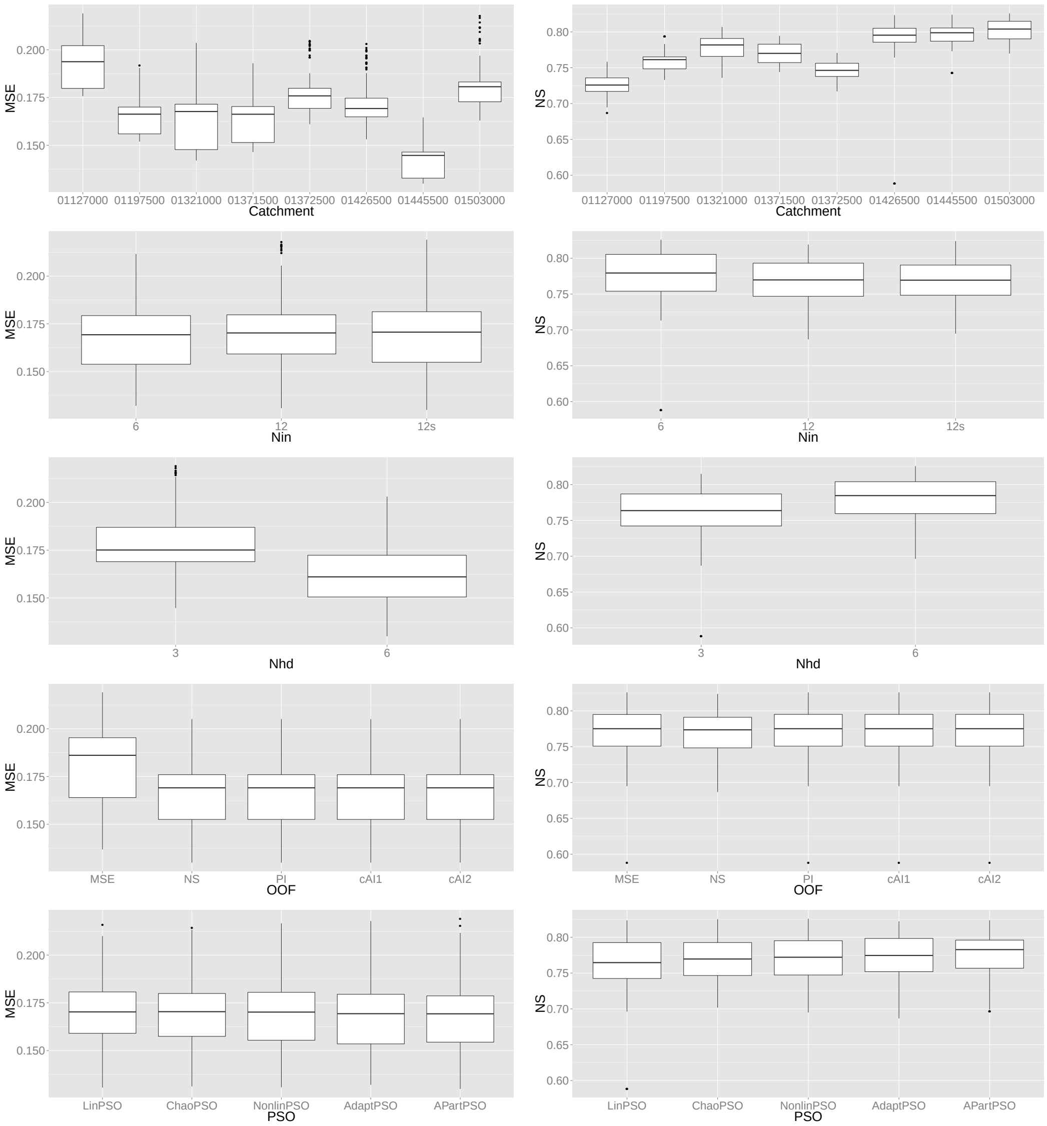

Figure 5.3: Box plots of all achieved values of MSE (left column) and NS (right column) during validation based on different factors

standard deviation of all achieved values for the best estimated hANN models. The results show that during calibration, lower MSE, $cAI1$ and $cAI2$, and higher NS were obtained. During validation, the PI criterion was higher, but still smaller than zero, which indicates that the naive model was better than hANN in many cases. The median of MSE for both hANN is approximately 1.42E-01, and median of NS is 8.29E-01, which indicates a good model fit.

Figures 5.4 and 5.5 present the time series of measured and simulated SPEI drought index. Models used for the visualization are the estimated best hANN for calibration and validation. The displayed catchments are the final best according to Table 5.5. It is evident, that the simulated SPEI is close to the measured one,

Table 5.5: The best levels of each factor for each accuracy criteria, and the final best level based on Tukey's HSD test. Minus sign indicates no significant difference in levels

Factor	**MSE**	**NS**	**PI**	**cAI1**	**cAI2**	**Final**
Calibration period						
Catch.	01371500	01371500	01197500	01197500	01371500	01371500
N_{in}	12, 12s	-	6	12, 12s	12, 12s	12, 12s
N_{hd}	6	6	6	6	6	6
OOF[a]	2, 3, 4, 5	-	1, 2, 4, 5	1, 2, 3, 5	1, 2, 3, 4	2
PSO[b]	-	5	5	-	-	5
Validation period						
Catch.	01445500	01503000	01127000	01372500	01445500	01445500
N_{in}	-	6	6	-	12, 12s	6
N_{hd}	6	6	6	6	-	6
OOF[a]	2, 3, 4, 5	-	-	1, 2, 3, 5	-	2, 3, 5
PSO[b]	-	4, 5	2, 3, 4, 5	-	-	4, 5

[a] $1 = MSE$, $2 = NS$, $3 = PI$, $4 = cAI1$, $5 = cAI2$

[b] $1 = LinPSO$, $2 = ChaoPSO$, $3 = NonlinPSO$, $4 = AdaptPSO$, $5 = APartPSO$

and that the models provide sufficient forecasts. The best hANN obtained during calibration provides good fit also for validation data, and vice versa. The only problem could be the overestimation of the lower values of SPEI.

The best hANN models for each catchment are displayed in Tables 5.7 and 5.8. They were estimated based on the performance according to NS objective function, and thus, this models provided the highest NS values. No statistical evaluation was performed within this analysis.

From the tables, it is evident that during calibration, only one model for each catchment gave the best results, except for the catchment with the USGS ID 01127000. On the other hand, more hANN models provided the same NS value for each catchment during validation, except for the catchment with the USGS ID 01321000. This is related to the statistical evaluation, based on which was determined that there is no significant difference between more levels of each factor during validation than during calibration.

5.4 Discussion

It was found out that the number of neurons in the hidden layer influenced results the most. In this research, the best choice for number of neurons in hidden layer

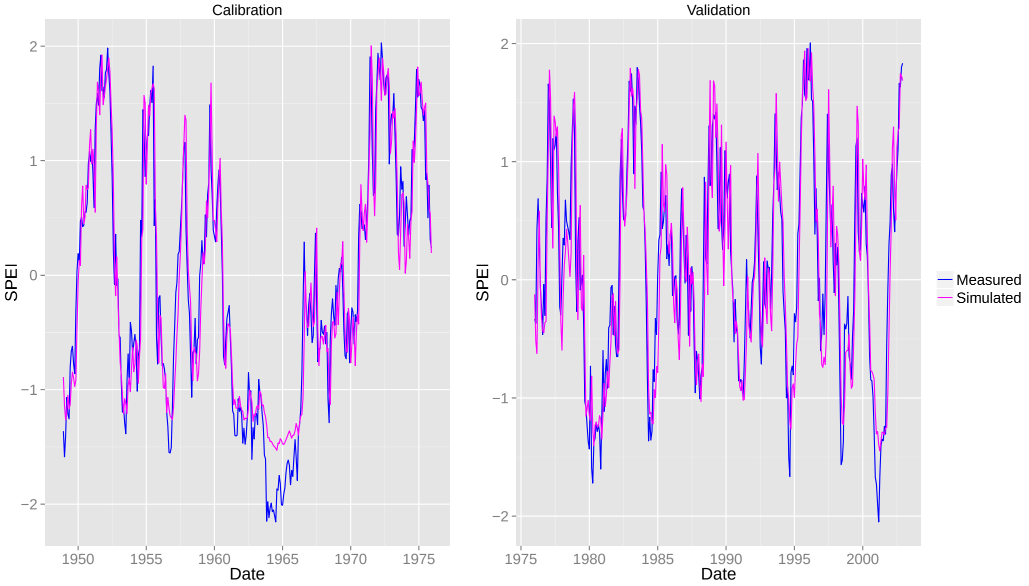

Figure 5.4: Measured and simulated time series of SPEI during calibration and validation period in the catchment 01371500 for the best hANN model 12-6-*APartPSO-NS*

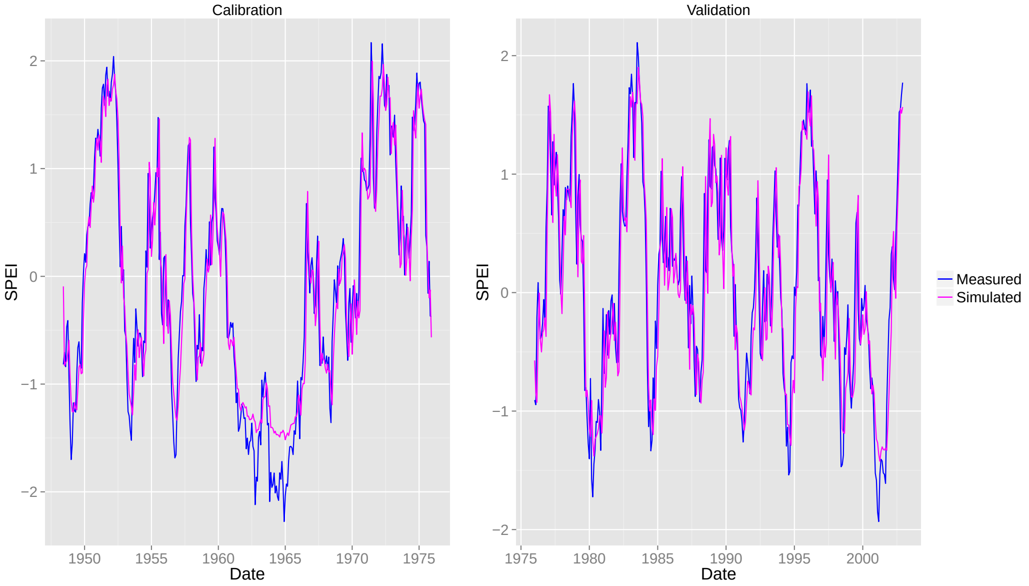

Figure 5.5: Measured and simulated time series of SPEI during calibration and validation period in the catchment 01445500 for the best hANN model 6-6-*AdaptPSO-NS*

Table 5.6: Statistical indices of the best hANN models

	MSE	**NS**	**PI**	**cAI1**	**cAI2**
Calibration period: 12-6-*APartPSO-NS*					
min	9.33E-02	7.30E-01	-9.83E-01	8.40E-01	3.08E-01
25%	1.08E-01	8.68E-01	-3.58E-01	9.14E-01	5.32E-01
median	1.14E-01	8.83E-01	-3.09E-01	9.99E-01	5.53E-01
75%	1.25E-01	8.91E-01	-2.55E-01	1.05E+00	5.82E-01
max	1.60E-01	9.06E-01	-8.04E-02	1.34E+00	6.50E-01
mean	1.16E-01	8.78E-01	-3.18E-01	9.95E-01	5.56E-01
sd	1.32E-02	1.86E-02	9.35E-02	9.71E-02	3.36E-02
Validation period: 6-6-*AdaptPSO-NS*					
min	1.30E-01	5.88E-01	-6.85E-01	8.83E-01	4.01E-01
25%	1.56E-01	7.50E-01	-2.82E-01	9.30E-01	6.67E-01
median	1.70E-01	7.75E-01	-2.16E-01	9.89E-01	6.95E-01
75%	1.80E-01	7.95E-01	-1.64E-01	1.02E+00	7.16E-01
max	2.19E-01	8.26E-01	-2.81E-02	1.24E+00	7.84E-01
mean	1.69E-01	7.71E-01	-2.29E-01	9.89E-01	6.92E-01
sd	1.82E-02	3.14E-02	8.95E-02	6.86E-02	3.81E-02

is equal to 6 for both calibration and validation period. It is in contrary with [139], who proposed $N_{hd} = log(T)$, where T is number of training samples. In this chapter, $T = 324$ for each calibration and validation, and thus, N_{hd} should be equal to 3. However, it was found out, that $N_{hd} = 3$ gave significantly worse results.

According to [82, 96], $N_{hd} = 2n + 1$, where n is number of input nodes. In the presented study, $n = 6$ or 12, and thus N_{hd} should be equal to 13 or 25. On the other hand, fewer neurons in hidden layer than in input layer gave good results [46, 155] while the number of search parameters is lower.

The best PSO variant for training the integrated hANN is the *APartPSO*, which gave significantly better results for both calibration and validation. This modification also provided good performance in optimization of benchmark functions as well as in parameter estimation during calibration of rainfall-runoff model [68]. Therefore, the optimization ability of the *APartPSO* method is suitable for solving real life engineering problems.

The best OOF in this research is the NS, which gave significantly better results for calibration and validation period. The Nash-Sutcliffe efficiency together with mean squared error criterion were used in many hydrological and meteorological studies, and are considered as suitable indicators of the model performance.

In this chapter, the best achieved MSE is equal to 0.093 and 0.130, the best achieved NS is equal to 0.909 and 0.826, during calibration and validation, respectively. The obtained results are superior than the research of [63]. In their study, the best achieved criteria for SPI simulations were MSE = 0.144 and NS = 0.834 for 12-month time scale computations.

On the other hand, better results were obtained by [8] during forecasting the SPI drought index through ANN models, where the results were MSE = 0.003 and NS = 0.953. Also [32] obtained better criteria during SPEI simulations, where MSE = 0.020 and NS = 0.983, but they used much more neurons in the hidden layer of the ANN, where N_{hd} = 43, which improved their performance.

As it was mentioned earlier in the text, the achieved PI values were usually less than zero. It indicates that the naive model was better than the hANN in many cases. Therefore, the PI was not considered as a suitable indicator, and for improving its performance, the number of generations within the PSO run should be increased. The $cAI1$ and $cAI2$ are accuracy criteria newly proposed in this chapter, and thus, their performances can not be compared with other studies.

Overall, in the forecasting of the SPEI drought index, the use of NS criteria as an optimized objective function is recommended. The choice of the PSO optimization method was not always essential for getting better results, but the adaptive method called *APartPSO* seems to be the best method out of the analysed variants. It is also important to note, that utilization of larger amount of neurons in the hidden layer of the ANN model can improve the performance, but the number of search parameters increases as does the computation time.

5.5 Conclusions

The main aim of the presented chapter was to combine hybrid neural network models with particle swarm optimization, which was used as training algorithm for the ANN weights. In total, 150 hybrid ANN models were applied for simulating the SPEI drought index on 8 US catchments. The dataset of 54 years of observations was divided into calibration and validation period, and the performance was analysed based on five measures of goodness of fit.

It was found out that the number of neurons in hidden layer of the ANN models influences results the most. Better performance was achieved with 6 neurons in the hidden layer instead of 3. The best number of neurons in the input layer was not determined uniquely. For calibration, better results were obtained with 12 inputs, compared to 6 input variables for validation.

Even though, the results obtained by different PSO variants were not always statistically different, the *APartPSO* is the most effective method for SPEI forecasting. The choice of PSO variant was not essential in all cases, but the adaptive variants gave better results in both calibration and validation.

The best objective function optimized by the final ANN model is the NS. In all cases, more OOF gave similar results, but in final evaluation of the model performances, the Nash-Sutcliffe efficiency was the most effective.

The results of this chapter extended the range of utilization of the particle swarm optimization technique and artificial neural network modelling. The combination of ANN with PSO is suitable for forecasting the SPEI drought index, and can be used for prediction of the potential threat of drought event.

The future studies should attempt to evaluate the sensitivity of each parameter of the hANN, and to explore influence of other network architecture, activation and output functions, or different optimized objective functions. The integrated ANN models are also promising for utilization in other real life engineering studies solving inverse problems.

Table 5.7: The best achieved accuracy criteria obtained with the best hANN models for each catchment during calibration period

Catchment	hANN	MSE	NS	PI	cAI1	cAI2
01127000	6-6-*AdaptPSO-MSE*	1.23E-01	8.78E-01	-2.82E-01	8.97E-01	6.00E-01
	6-6-*AdaptPSO-PI*	1.13E-01	8.78E-01	-2.93E-01	8.97E-01	6.00E-01
	6-6-*AdaptPSO-cAI*1	1.13E-01	8.78E-01	-2.82E-01	9.86E-01	6.00E-01
	6-6-*AdaptPSO-cAI*2	1.13E-01	8.78E-01	-2.82E-01	8.97E-01	6.30E-01
01197500	12*s*-6-*APartPSO-NS*	9.64E-02	8.98E-01	-2.43E-01	8.47E-01	5.16E-01
01321000	6-6-*AdaptPSO-NS*	9.42E-02	8.93E-01	-2.25E-01	8.70E-01	5.37E-01
01371500	12*s*-6-*APartPSO-NS*	9.42E-02	9.09E-01	-2.55E-01	8.47E-01	4.96E-01
01372500	12*s*-6-*NonlinPSO-NS*	1.08E-01	8.98E-01	-3.17E-01	9.30E-01	5.38E-01
01426500	12*s*-6-*APartPSO-NS*	1.05E-01	8.83E-01	-2.65E-01	8.89E-01	5.43E-01
01445500	12*s*-6-*APartPSO-NS*	9.85E-02	9.02E-01	-1.74E-01	8.68E-01	5.13E-01
01503000	6-3-*APartPSO-MSE*	1.28E-01	8.67E-01	-8.04E-02	9.15E-01	3.08E-01

Table 5.8: The best achieved accuracy criteria obtained with the best hANN models for each catchment during validation period

Catchment	**hANN**	**MSE**	**NS**	**PI**	**cAI1**	**cAI2**
01127000	6-6-*AdaptPSO-MSE*	1.76E-01	7.58E-01	-1.78E-01	8.96E-01	7.46E-01
	6-6-*AdaptPSO-PI*	1.76E-01	7.58E-01	-1.18E-01	8.96E-01	7.46E-01
	6-6-*AdaptPSO-cAI*1	1.76E-01	7.58E-01	-1.78E-01	9.07E-01	7.46E-01
	6-6-*AdaptPSO-cAI*2	1.76E-01	7.58E-01	-1.78E-01	8.96E-01	7.29E-01
01197500	12-6-*APartPSO-MSE*	1.65E-01	7.94E-01	-1.36E-01	9.30E-01	6.83E-01
	12-6-*APartPSO-PI*	1.61E-01	7.94E-01	-2.31E-01	9.30E-01	6.83E-01
	12-6-*APartPSO-cAI*1	1.61E-01	7.94E-01	-1.36E-01	9.98E-01	6.83E-01
	12-6-*APartPSO-cAI*2	1.61E-01	7.94E-01	-1.36E-01	9.30E-01	7.01E-01
01321000	12*s*-6-*LinPSO-NS*	1.45E-01	8.07E-01	-3.08E-01	9.04E-01	6.48E-01
01371500	6-6-*APartPSO-MSE*	1.62E-01	7.95E-01	-1.27E-01	9.25E-01	6.99E-01
	6-6-*APartPSO-PI*	1.51E-01	7.95E-01	-1.36E-01	9.25E-01	6.99E-01
	6-6-*APartPSO-cAI*1	1.51E-01	7.95E-01	-1.27E-01	9.65E-01	6.99E-01
	6-6-*APartPSO-cAI*2	1.51E-01	7.95E-01	-1.27E-01	9.25E-01	6.87E-01
01372500	12*s*-6-*APartPSO-MSE*	1.76E-01	7.71E-01	-2.06E-01	9.15E-01	7.16E-01
	12*s*-6-*APartPSO-PI*	1.64E-01	7.71E-01	-1.76E-01	9.15E-01	7.16E-01
	12*s*-6-*APartPSO-cAI*1	1.64E-01	7.71E-01	-2.06E-01	9.74E-01	7.16E-01
	12*s*-6-*APartPSO-cAI*2	1.64E-01	7.71E-01	-2.06E-01	9.15E-01	7.40E-01
01426500	6-6-*NonlinPSO-MSE*	1.72E-01	8.23E-01	-2.81E-02	9.22E-01	6.87E-01
	6-6-*NonlinPSO-PI*	1.56E-01	8.23E-01	-2.14E-01	9.22E-01	6.87E-01
	6-6-*NonlinPSO-cAI*1	1.56E-01	8.23E-01	-2.81E-02	9.91E-01	6.87E-01
	6-6-*NonlinPSO-cAI*2	1.56E-01	8.23E-01	-2.81E-02	9.22E-01	7.18E-01
01445500	12*s*-6-*ChaoPSO-MSE*	1.38E-01	8.24E-01	-1.45E-01	9.23E-01	6.20E-01
	12*s*-6-*ChaoPSO-PI*	1.31E-01	8.24E-01	-2.00E-01	9.23E-01	6.20E-01
	12*s*-6-*ChaoPSO-cAI*1	1.31E-01	8.24E-01	-1.45E-01	9.62E-01	6.20E-01
	12*s*-6-*ChaoPSO-cAI*2	1.31E-01	8.24E-01	-1.45E-01	9.23E-01	6.29E-01
01503000	6-6-*NonlinPSO-MSE*	1.77E-01	8.26E-01	-1.52E-01	9.28E-01	7.05E-01
	6-6-*NonlinPSO-PI*	1.64E-01	8.26E-01	-1.53E-01	9.28E-01	7.05E-01
	6-6-*NonlinPSO-cAI*1	1.64E-01	8.26E-01	-1.52E-01	9.63E-01	7.05E-01
	6-6-*NonlinPSO-cAI*2	1.64E-01	8.26E-01	-1.52E-01	9.28E-01	6.97E-01

CHAPTER 6

PRINCIPAL CONCLUSIONS AND SUMMARY

Finding the optimal state of reality is the main purpose of the optimization process. The best variant from many possibilities is selected, and the effectiveness of the given system increases. Optimization has been applied in many real life engineering problems as in hydrological modelling. Within the hydrological case studies, the optimization process serves to estimate the best set of model parameters, or to train model weights in artificial neural networks.

Due to difficulties, which may occur during optimization, it is necessary to wisely choose a suitable method. Based on the optimization problem, it is recommended to devote some time modifying the selected optimization method.

In this doctoral thesis, I focused on the particle swarm optimization technique, and its utilization in hydrological modelling. It is relatively recent optimization method, which has only a few parameters to adjust, and is easy to implement to the selected problem. The original algorithm was modified by many authors. They focused on changing the initialization of particles in the swarm, updating the population topology, adding new parameters into the equation, or incorporating shuffling mechanism into the algorithm.

The main goals of the thesis were provision of comprehensive review about the PSO method, implementation of selected PSO modifications together with a new proposed variant in C++ programming language, and application of the best modifications in real-life optimization problems from the field of hydrology.

The comprehensive review about the PSO technique was provided in Chapter 2. Due to the limited space in the thesis, I focused mainly on features, which were thereafter useful for my research. The original equations with different modifications were summarized there together with various topologies and applicable objective functions.

Comparison of selected PSO modifications was provided in Chapter 3. In total, 27 modifications were tested on 5 uni-modal and 6 multi-modal benchmark problems. Variants with constriction factor and different types of inertia weight were analysed. The results showed that the best PSO variant is the method with adaptive inertia weight parameter. In addition, the shuffled complex evolution strategy improved the performance, and gave the best results, which confirmed the usefulness of this approach. Therefore, I decided to later focus the attention to this direction of possible modifications, i.e. adaptive version of inertia weight, and sub-swarms with shuffling and redistribution of particles.

In Chapter 4, a new PSO variant was proposed. The method enhances the global exploration and local exploitation in the parametric space during the optimization process through new adaptive strategy of inertia weight. The shuffled complex evolution strategy was incorporated into the algorithm. The optimization ability of the proposed method was tested on 11 benchmark problems, and the obtained results were compared with 3 PSO modifications from Chapter 3. It was found out that the new proposed variant performs well, and has suitable results.

Due to the fact, that the new proposed PSO version achieved good results in optimizing benchmark functions, it was applied in two real-life optimization problems. One case study concerned with hydrological model Bilan (Chapter 4), and second case study dealt with artificial neural networks (Chapter 5).

The new method together with other 3 PSO modifications was used for finding the solution of inverse problems related to estimation of parameters of rainfall-runoff model Bilan (Chapter 4). Based on statistical tests, it was concluded that the best results were obtained by the new proposed method and by the adaptive variant, which was also the best method in Chapter 3. On the other hand, the PSO modification with parameter of constriction factor performed the worst, which is also in agreement with the findings of Chapter 3.

The 4 best PSO modifications from Chapter 3 together with the proposed method from Chapter 4 were combined with artificial neural networks in Chapter 5. The integrated hybrid models were used for forecasting the standardized precipitation evapotranspiration drought index. The influence of each PSO method and other variables on the simulations was analysed. The variable, which influenced the results the most, was number of neurons in hidden layer of the ANN models. Therefore, it is essential to choose the size of hidden layer appropriately. In terms of PSO method, the most effective technique for SPEI forecasting was the proposed variant from Chapter 4.

Based on the results obtained during my research, I can conclude that adaptive version of inertia weight parameter is the most effective approach from all analysed variants. The shuffled complex evolution also significantly improves the optimization. The new PSO method proposed in this thesis finds the optimum value not only in benchmark problems, but also in real-life optimization problems. Therefore, it can be applied in other engineering studies.

Overall, the contribution of the doctoral thesis for the current stage of scientific knowledge is evident from the individual chapters. The results of this thesis extended the utilization of PSO methods in real-life engineering optimization problems. All analysed PSO algorithms are available for later use, and the completed algorithms are basis for other research projects.

CHAPTER 7

SHRNUTÍ

Hlavním cílem optimalizačního procesu je nalezení optimálního stavu dané reality. Z mnoha možností je vybrána nejlepší varianta, čímž vzroste efektivita celého systému. Optimalizační technika byla aplikována v mnoha inženýrských problémech. V rámci hydrologického modelování je využita k odhadu nejlepší sady parametrů modelu, či k trénování umělých neuronových sítí.

Relativně novou optimalizační metodou je optimalizace rojem částic (PSO), která se vyznačuje malým množstvím parametrů pro nastavení a jednoduchou implementací. Původní algoritmus této metody byl mnoha autory modifikován. Důraz byl kladen na změnu způsobu inicializace částic v hejnu, aktualizaci topologie populace, přidání nového parametru do rovnice, či začlenění mechanismu promíchávání do algoritmu.

Modifikace PSO algoritmu zlepší provedení optimalizace, zamezí předčasné konvergenci a sníží výpočetní čas systému. Z těchto důvodů zahrnují hlavní cíle předložené doktorské práce navržení nové modifikace PSO metody s její implementací v programovacím jazyce C++. V práci bylo porovnáno a vyhodnoceno více PSO variant a nejlepší metody byly použity ve dvou hydrologických případových studiích.

První případová studie se zabývá použitím PSO algoritmů na inverzních problémech spojených s odhadem parametrů srážko-odtokového modelu Bilan. Ve druhé studii byly zkombinovány umělé neuronové sítě s PSO metodou pro předpověď vybraného indexu sucha.

Bylo zjištěno, že optimalizace rojem částic je vhodným nástrojem pro řešení problémů v rámci hydrologického modelování. Nejefektivnějšími PSO modifikacemi jsou varianty s adaptivní verzí váhovacího faktoru, které aktualizují rychlost částice během prohledávání vícedimenzionální řešené oblasti pomocí zpětné vazby. Mechanismus promíchávání a přerozdělování částic do komplexů, ve kterých je samostatně spouštěn PSO algoritmus, také výrazně zlepšil provedení optimalizace.

Přínos této doktorské práce spočívá ve vytvoření nové PSO modifikace, která byla otestována na referenčních problémech a úspěšně aplikována ve dvou hydrologických případových studiích. Výsledky práce rozšířily využití PSO metody v reálných inženýrských problémech a všechny analyzované PSO algoritmy jsou k dispozici pro pozdější využití v rámci dalších výzkumných projektů.

Bibliography

[1] Angeline, P. J. Using selection to improve particle swarm optimization. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation* (Anchorage, AK, 1998), pp. 84–89.

[2] Arsenault, R., Poulin, A., Cote, P., and Brissette, F. Comparison of Stochastic Optimization Algorithms in Hydrological Model Calibration. *Journal of hydrologic engineering 19*, 7 (2014), 1374–1384.

[3] Arumugam, M. S., and Rao, M. V. C. On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems. *Applied Soft Computing 8* (2008), 324–336.

[4] Baltar, A. M., and Fontane, D. G. Use of multiobjective particle swarm optimization in water resources management. *Journal of water resources planning and management-ASCE 134*, 3 (2008), 257–265.

[5] Bansal, J. C., Singh, P. K., Saraswat, M., Verma, A., Jadon, S. S., and Abraham, A. Inertia Weight strategies in Particle Swarm Optimization. In *Third World Congress on Nature and Biologically Inspired Computing (NaBIC)* (Salamaca, 2011), pp. 640–647.

[6] Beguería, S., and Vicente-Serrano, S. M. *SPEI: Calculation of the Standardised Precipitation-Evapotranspiration Index*, 2013. R package version 1.6.

[7] Beguería, S., Vicente-Serrano, S. M., Reig, F., and Latorre, B. Standardized precipitation evapotranspiration index (SPEI) revisited: parameter fitting, evapotranspiration models, tools, datasets and drought monitoring. *International Journal of Climatology 34*, 10 (2014), 3001–3023.

[8] BELAYNEH, A., AND ADAMOWSKI, J. Standard precipitation index drought forecasting using neural networks, wavelet neural networks, and support vector regression. *Applied Computational Intelligence and Soft Computing 2012* (2012), 6.

[9] BENNETT, N. D., CROKE, B. F. W., GUARISO, G., GUILLAUME, J. H. A., HAMILTON, S. H., JAKEMAN, A. J., MARSILI-LIBELLI, S., NEWHAM, L. T. H., NORTON, J. P., PERRIN, C., PIERCE, S. A., ROBSON, B., SEPPELT, R., VOINOV, A. A., FATH, B. D., AND ANDREASSIAN, V. Characterising performance of environmental models. *Environmental modelling & software 40* (2013), 1–20.

[10] BERGH, F. V. D. *An Analysis of Particle Swarm Optimizers*. PhD thesis, University of Pretoria, 2001.

[11] BERGH, F. V. D., AND ENGELBRECHT, A. P. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation 8*, 3 (2004), 225–239.

[12] BLACKWELL, T., AND BRANKE, J. Multiswarms, exclusion, and anti-convergence in dynamic environments. *Evolutionary Computation, IEEE Transactions on 10*, 4 (2006), 459–472.

[13] BLUM, C. Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews 2*, 4 (2005), 353–373.

[14] BLUM, C., AND MERKLE, D. *Swarm Intelligence: Introduction and Application*. Springer, 2008.

[15] BOYLE, D., GUPTA, H., AND SOROOSHIAN, S. Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods. *Water resources research 36*, 12 (2000), 3663–3674.

[16] BRAUER, C. C., TEULING, A. J., TORFS, P. J. J. F., AND UIJLENHOET, R. The Wageningen Lowland Runoff Simulator (WALRUS): a lumped rainfall-runoff model for catchments with shallow groundwater. *Geoscientific Model Development 7*, 5 (2014), 2313–2332.

[17] BRAUER, C. C., TORFS, P. J. J. F., TEULING, A. J., AND UIJLENHOET, R. The Wageningen Lowland Runoff Simulator (WALRUS): application to the Hupsel Brook catchment and the Cabauw polder. *Hydrology and Earth System Sciences 18*, 10 (2014), 4007–4028.

[18] BUI, L. T., SOLIMAN, O., AND ABBASS, H. A. A modified strategy for the constriction factor in particle swarm optimization. In *Proceedings of the 3rd Australian conference on Progress in artificial life* (Berlin, Heidelberg, 2007), ACAL'07, Springer-Verlag, pp. 333–344.

[19] CABRERA, J. C. F., AND COELLO, C. A. C. Handling constraints in particle swarm optimization using a small population size. In *Proceedings of the artificial intelligence 6th Mexican international conference on Advances in artificial intelligence* (Berlin, Heidelberg, 2007), MICAI'07, Springer-Verlag, pp. 41–51.

[20] CHAU, K. W. Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *Journal of hydrology 329*, 3 (2006), 363–367.

[21] CHAU, K. W. A split-step particle swarm optimization algorithm in river stage forecasting. *Journal of hydrology 346*, 3 (2007), 131–135.

[22] CHIEW, F., AND MCMAHON, T. Assessing the adequacy of catchment streamflow yield estimates. *Soil Research 31*, 5 (1993), 665–680.

[23] CHOU, C.-M. Particle Swarm Optimization for Identifying Rainfall-Runoff Relationships. *Journal of Water Resource and Protection 4*, 3 (2012), 115–126.

[24] CLERC, M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington, DC, 1999), pp. 1951–1957.

[25] CLERC, M. Back to random topology. Tech. rep., Particle Swarm Central, 2007.

[26] CLERC, M. Standard Particle Swarm Optimisation. Tech. rep., Particle Swarm Central, 2012.

[27] CLERC, M., AND KENNEDY, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation 6*, 1 (2002), 58–73.

[28] CORNE, D., DORIGO, M., GLOVER, F., DASGUPTA, D., MOSCATO, P., POLI, R., AND PRICE, K. V., Eds. *New ideas in optimization*. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.

[29] Dawson, C. W., Abrahart, R. J., and See, L. M. HydroTest: A web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts. *Environmental Modelling and Software 22*, 7 (2007), 1034–1052.

[30] Dawson, C. W., Abrahart, R. J., and See, L. M. Hydro Test: Further development of a web resource for the standardised assessment of hydrological models. *Environmental modelling & software 25*, 11 (2010), 1481–1482.

[31] Deng, W., Chen, R., He, B., Liu, Y., Yin, L., and Guo, J. A novel two-stage hybrid swarm intelligence optimization algorithm and application. *Soft Comput. 16*, 10 (2012), 1707–1722.

[32] Deo, R. C., and Şahin, M. Application of the Artificial Neural Network model for prediction of monthly Standardized Precipitation and Evapotranspiration Index using hydrometeorological parameters and climate indices in eastern Australia. *Atmospheric Research 161-162*, 1 (2015), 65–81.

[33] Derrac, J., García, S., Molina, D., and Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation 1*, 1 (2011), 3–18.

[34] Duan, Q., Gupta, V., and Sorooshian, S. Shuffled complex evolution approach for effective and efficient global minimization. *Journal of Optimization Theory and Applications 76*, 3 (1993), 501–521.

[35] Duan, Q., Schaake, J., Andreassian, V., Franks, S., Goteti, G., Gupta, H., Gusev, Y., Habets, F., Hall, A., Hay, L., et al. Model Parameter Estimation Experiment (MOPEX): An overview of science strategy and major results from the second and third workshops. *Journal of Hydrology 320*, 1 (2006), 3–17.

[36] Duan, Q., Sorooshian, S., and Gupta, V. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research 28*, 4 (1993), 1015–1031.

[37] Duch, W., and Jankowski, N. Survey of neural transfer functions. *Neural Computing Surveys 2*, 1 (1999), 163–212.

[38] DURSTENFELD, R. Algorithm 235: Random permutation. *Commun. ACM 7*, 7 (1964), 420–.

[39] DYTHAM, C. *Choosing and using statistics: a biologist's guide*. Blackwell Science Ltd., 2011.

[40] EBERHART, R., AND SHI, Y. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation* (La Jolla, CA, 2000), vol. 1(7), pp. 84–88.

[41] EBERHART, R., AND SHI, Y. Particle swarm optimization: developments, applications and resources. In *Proceedings of the 2001 Congress on Evolutionary Computation* (Seoul, 2001), vol. 1, pp. 81–86.

[42] EBERHART, R., AND SHI, Y. Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the 2001 Congress on Evolutionary Computation* (2001), vol. 1, pp. 94–100.

[43] EBERHART, R., AND SHI, Y. *Computational Intelligence: Concepts to Implementations*. Morgan Kaufmann, 2007.

[44] EBERHART, R., SIMPSON, P., AND DOBBINS, R. *Computational intelligence PC tools*. Academic Press Professional, Inc., San Diego, CA, USA, 1996.

[45] FENG, Y., TENG, G.-F., WANG, A.-X., AND YAO, Y.-M. Chaotic inertia weight in particle swarm optimization. In *Second International Conference on Innovative Computing, Information and Control* (Kumamoto, 2007), p. 475.

[46] FLETCHER, D., AND GOSS, E. Forecasting with neural networks: an application using bankruptcy data. *Information & Management 24*, 3 (1993), 159–167.

[47] GARCÍA, S., MOLINA, D., LOZANO, M., AND HERRERA, F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics 15*, 6 (2009), 617–644.

[48] GAUR, S., CHAHAR, B. R., AND GRAILLOT, D. Analytic elements method and particle swarm optimization based simulation–optimization model for groundwater management. *Journal of Hydrology 402*, 3 (2011), 217–227.

[49] GERSHENFELD, N. *The nature of mathematical modeling*. Cambridge University Press, 1999.

[50] GILL, M. K., KAHEIL, Y. H., KHALIL, A., MCKEE, M., AND BASTIDAS, L. Multiobjective particle swarm optimization for parameter estimation in hydrology. *Water Resources Research 42*, 7 (2006).

[51] GIMMLER, J., STÜTZLE, T., AND EXNER, T. E. Hybrid particle swarm optimization: an examination of the influence of iterative improvement algorithms on performance. In *Proceedings of the 5th international conference on Ant Colony Optimization and Swarm Intelligence* (Berlin, Heidelberg, 2006), ANTS'06, Springer-Verlag, pp. 436–443.

[52] GOSWAMI, M., O'CONNOR, K., BHATTARAI, K., AND SHAMSELDIN, A. Assessing the performance of eight real-time updating models and procedures for the Brosna River. *Hydrology and Earth System Sciences Discussions 9*, 4 (2005), 394–411.

[53] HANEL, M., MRKVIČKOVA, M., MÁCA, P., VIZINA, A., AND PECH, P. Evaluation of Simple Statistical Downscaling Methods for Monthly Regional Climate Model Simulations with Respect to the Estimated Changes in Runoff in the Czech Republic. *Water Resources Management 27*, 15 (2013), 5261–5279.

[54] HANEL, M., VIZINA, A., MÁCA, P., AND PAVLÁSEK, J. A Multi-model Assessment of Climate Change Impact on Hydrological Regime in the Czech Republic. *Journal of Hydrology and Hydromechanics 60*, 3 (2012), 152–161.

[55] HANSEN, N. Compilation of results on the 2005 CEC benchmark function set. *Online, May* (2006).

[56] HASLINGER, K., KOFFLER, D., SCHÖNER, W., AND LAAHA, G. Exploring the link between meteorological drought and streamflow: Effects of climate-catchment interaction. *Water Resources Research 50*, 3 (2014), 2468–2487.

[57] HASSAN, R., COHANIM, B., WECK, O. D., AND VENTER, G. A comparison of particle swarm optimization and the genetic algorithm. In *1st AIAA multidisciplinary design optimization specialist conference* (2005), pp. 1–13.

[58] HAYES, M., SVOBODA, M., WALL, N., AND WIDHALM, M. The Lincoln declaration on drought indices: universal meteorological drought index

recommended. *Bulletin of the American Meteorological Society 92*, 4 (2011), 485–488.

[59] HE, X., GUAN, H., AND QIN, J. A hybrid wavelet neural network model with mutual information and particle swarm optimization for forecasting monthly rainfall. *Journal of Hydrology 527* (2015), 88–100.

[60] HERNANDEZ, E. A., AND UDDAMERI, V. Standardized precipitation evaporation index (SPEI)-based drought assessment in semi-arid south Texas. *Environmental earth sciences 71*, 6 (2014), 2491–2501.

[61] HORÁČEK, S., KAŠPÁREK, L., AND NOVICKÝ, O. Estimation of climate change impact on water resources by using Bilan water balance model. In *IOP conference series: earth and environmental science* (2008), vol. 4, IOP Publishing, p. 012023.

[62] HORNIK, K., STINCHCOMBE, M., AND WHITE, H. Multilayer feedforward networks are universal approximators. *Neural networks 2*, 5 (1989), 359–366.

[63] HOSSEINI-MOGHARI, S., AND ARAGHINEJAD, S. Monthly and seasonal drought forecasting using statistical neural networks. *Environmental Earth Sciences 74*, 1 (2015), 397–412.

[64] HUO, Z., FENG, S., KANG, S., HUANG, G., WANG, F., AND GUO, P. Integrated neural networks for monthly river flow estimation in arid inland basin of Northwest China. *Journal of Hydrology 420* (2012), 159–170.

[65] IMRAN, M., JABEEN, H., AHMAD, M., ABBAS, Q., AND BANGYAL, W. Opposition based PSO and mutation operators. In *Education Technology and Computer (ICETC), 2010 2nd International Conference on* (2010), vol. 4, IEEE, pp. V4–506.

[66] JABEEN, H., JALIL, Z., AND BAIG, A. R. Opposition based initialization in particle swarm optimization (O-PSO). In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers* (2009), ACM, pp. 2047–2052.

[67] JAKUBCOVÁ, M., MÁCA, P., AND PECH, P. A comparison of selected modifications of the particle swarm optimization algorithm. *Journal of Applied Mathematics 2014* (2014), 1–10.

[68] JAKUBCOVÁ, M., MÁCA, P., AND PECH, P. Parameter estimation in rainfall-runoff modelling using distributed versions of particle swarm optimization algorithm. *Mathematical Problems in Engineering 2015* (2015), 1–13.

[69] JIANG, Y., HU, T., HUANG, C., AND WU, X. An improved particle swarm optimization algorithm. *Applied Mathematics and Computation 193*, 1 (2007), 231–239.

[70] JIANG, Y., LI, X., AND HUANG, C. Automatic calibration a hydrological model using a master-slave swarms shuffling evolution algorithm based on self-adaptive particle swarm optimization. *Expert Systems with Applications: An International Journal 40*, 2 (2013), 752–757.

[71] KARABOGA, D. An idea based on honey bee swarm for numerical optimization. Tech. rep., Technical report-tr06, Erciyes university, Engineering faculty, Computer engineering department, 2005.

[72] KAŠPÁREK, L., HANEL, M., HORÁČEK, S., MÁCA, P., AND VIZINA, A. *bilan: Bilan water balance model*, 2014. T. G. Masaryk Water Research Institute and p.r.i., R package version 2013.12.

[73] KENNEDY, J., AND EBERHART, R. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks* (Perth, WA, 1995), pp. 1942–1948.

[74] KENNEDY, J., EBERHART, R., AND SHI, Y. *Swarm Intelligence*. Morgan Kaufmann, 2001.

[75] KENNEDY, J., AND MENDES, R. Population structure and particle swarm performance. In *Proceedings of the Congress on Evolutionary Computation (CEC 2002)* (2002), vol. 2, IEEE Press, pp. 1671–1676.

[76] KENNEDY, J., AND MENDES, R. Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms. In *Proceedings of the 2003 IEEE SMC Workshop on Soft Computing in Industrial Applications (SMCia03)* (2003), IEEE Computer Society, pp. 45–50.

[77] KRISHNANAND, K., AND GHOSE, D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE* (2005), IEEE, pp. 84–91.

[78] LANEN VAN, H., TALLAKSEN, L., KAŠPÁREK, L., AND QUERNER, E. Hydrological drought analysis in the Hupsel basin using different physically-based models. In *FRIEND'97-Regional Hydrology: Concepts and Models for Sustainable Water Resource Management* (1997), A. Gustard and al., Eds., no. 246, UNESCO; WMO; EC; IAHS; MST Slovenia, pp. 189–196.

[79] LAWRENCE, D., HADDELAND, I., AND LANGSHOLT, E. Calibration of HBV hydrological models using PEST parameter estimation. *Oslo, Norway: Norwegian Water Resources and Energy Directorate* (2009).

[80] LIANG, J. J., AND SUGANTHAN, P. N. Dynamic multi-swarm particle swarm optimizer with local search. In *The 2005 IEEE Congress on Evolutionary Computation* (Edinburgh, Scotland, 2005), vol. 1(1), pp. 522 – 528.

[81] LIM, C. P., JAIN, L. C., AND DEHURI, S. *Innovations in swarm intelligence*. Springer, 2009.

[82] LIPPMANN, R. P. An introduction to computing with neural nets. *ASSP Magazine, IEEE 4*, 2 (1987), 4–22.

[83] LIU, F., ZHOU, J.-Z., FANG, R.-C., PENG, B., AND YANG, J.-J. An improved particle swarm optimization and its application in long-term streamflow forecast. In *Proceedings of the 2005 International Conference on Machine Learning and Cybernetics* (Guangzhou, China, 2005), pp. 2913–2918.

[84] LÜ, H., YU, Z., HORTON, R., ZHU, Y., WANG, Z., HAO, Z., AND XIANG, L. Multi-scale assimilation of root zone soil water predictions. *Hydrological Processes 25*, 20 (2011), 3158–3172.

[85] LUKE, S. Essentials of Metaheuristics. `http://cs.gmu.edu/~sean/book/metaheuristics/Essentials.pdf`, 2011. Online, accessed 10. 8. 2015.

[86] MÁCA, P., PECH, P., AND PAVLÁSEK, J. Comparing the Selected Transfer Functions and Local Optimization Methods for Neural Network Flood Runoff Forecast. *Mathematical Problems in Engineering* (2014), 1–10.

[87] MÁCA, P., VIZINA, A., AND HORÁČEK, S. The parameter optimisation of hydrological model BILAN using the SCDE method (in Czech). *Vodohospodářské technicko-ekonomické informace* (2013), 1–4.

[88] MACHLICA, A., HORVAT, O., HORÁČEK, S., OOSTERWIJK, J., VAN LOON, A. F., FENDEKOVA, M., AND VAN LANEN, H. A. J. Influence of Model

Structure on Base Flow Estimation Using Bilan, FRIER and HBV-LIGHT Models. *Journal of Hydrology and Hydromechanics 60*, 4 (2012), 242–251.

[89] MARIANI, V. C., GUILHERME, L., LUVIZOTTO, J., GUERRA, F. A., AND COELHO, L. D. S. A hybrid shuffled complex evolution approach based on differential evolution for unconstrained optimization. *Applied mathematics and computation 217*, 12 (2011), 5822–5829.

[90] MAXWELL, I. Processing: Autonomous Steering Behaviours - Part 02. `http://www.supermanoeuvre.com/blog/?p=327`, 2009. Online, accessed 20. 5. 2013.

[91] MCKAY, M. D., BECKMAN, R. J., AND CONOVER, W. J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 21*, 2 (1979), 239–245.

[92] MCKEE, T. B., DOESKEN, N. J., KLEIST, J., ET AL. The relationship of drought frequency and duration to time scales. In *Proceedings of the 8th Conference on Applied Climatology* (1993), vol. 17, American Meteorological Society Boston, MA, USA, pp. 179–183.

[93] MENDES, R. *Population topologies and their influence in particle swarm performance*. PhD thesis, University of Minho, 2004.

[94] MICHALEWICZ, Z., AND FOGEL, D. *How to Solve It: Modern Heuristics*. Springer, 2004.

[95] MILLONAS, M. M. Swarms, phase transitions and collective intelligence. In *Artificial Life III*, C. Langton, Ed. Addison-Wesley, 1994.

[96] MISHRA, A., AND DESAI, V. Drought forecasting using feed-forward recursive neural network. *Ecological Modelling 198*, 1 (2006), 127–138.

[97] MOLINA, D., HERRERA, F., AND LOZANO, M. Adaptive local search parameters for real-coded memetic algorithms. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on* (2005), vol. 1, IEEE, pp. 888–895.

[98] MOSSAD, A., AND ALAZBA, A. A. Drought Forecasting Using Stochastic Models in a Hyper-Arid Climate. *Atmosphere 6*, 4 (2015), 410–430.

[99] MPELASOKA, F., HENNESSY, K., JONES, R., AND BATES, B. Comparison of suitable drought indices for climate change impacts assessment over Australia towards resource management. *International Journal of Climatology 28*, 10 (2008), 1283–1292.

[100] MUNOZ, D. M., LLANOS, C. H., COELHO, L. D. S., AND AYALA-RINCON, M. Opposition-based Shuffled PSO with Passive Congregation Applied to FM Matching Synthesis. In *2011 IEEE Congress on Evolutionary Computation (CEC)* (2011), IEEE Congress on Evolutionary Computation, pp. 2775–2781.

[101] NASH, J. E., AND SUTCLIFFE, J. V. River flow forecasting through conceptual models part I - A discussion of principles. *Journal of hydrology 10*, 3 (1970), 282–290.

[102] NICKABADI, A., EBADZADEH, M. M., AND SAFABAKHSH, R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing 11*, 4 (2011), 3658–3670.

[103] NIU, B., ZHU, Y., HE, X., AND WU, H. MCPSO: A multi-swarm cooperative particle swarm optimizer. *Applied Mathematics and Computation 185*, 2 (2007), 1050–1062.

[104] NOURANI, V., AND KALANTARI, O. Integrated artificial neural network for spatiotemporal modeling of rainfall–runoff–sediment processes. *Environmental Engineering Science 27*, 5 (2010), 411–422.

[105] OTT, R., AND LONGNECKER, M. *An introduction to statistical methods and data analysis*. Cengage Learning, 2008.

[106] PALMER, W. C. *Meteorological drought*, vol. 30. US Department of Commerce, Weather Bureau Washington, DC, USA, 1965.

[107] PANIGRAHI, B. K., PANDI, V. R., AND DAS, S. Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. *Energy conversion and management 49*, 6 (2008), 1407–1415.

[108] PANT, M., THANGARAJ, R., AND ABRAHAM, A. Particle swarm optimization using adaptive mutation. In *Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on* (2008), IEEE, pp. 519–523.

[109] PANT, M., THANGARAJ, R., AND ABRAHAM, A. Particle swarm optimization: Performance tuning and empirical analysis. In *Foundations of Computational Intelligence Volume 3*, A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht, Eds., vol. 203 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, 2009, pp. 101–128.

[110] PANT, M., THANGARAJ, R., GROSAN, C., AND ABRAHAM, A. Improved particle swarm optimization with low-discrepancy sequences. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on* (2008), IEEE, pp. 3011–3018.

[111] PARSOPOULOS, K. E., PLAGIANAKOS, V. P., MAGOULAS, G. D., AND VRAHATIS, M. N. *Advances in Convex Analysis and Global Optimization*. Springer, 2001, ch. Improving particle swarm optimizer by function stretching, pp. 445–457.

[112] PARSOPOULOS, K. E., AND VRAHATIS, M. N. *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*. WSEAS Press, 2002, ch. Initializing the particle swarm optimizer using the nonlinear simplex method, pp. 216–221.

[113] PARSOPOULOS, K. E., AND VRAHATIS, M. N. On the computation of all global minimizers through particle swarm optimization. *IEEE Transactions on Evolutionary Computation 8*, 3 (2004), 211–224.

[114] PAULING, L. *The nature of the chemical bond*. Cornell, Univ. Press, 1939.

[115] PEREIRA, L. S., CORDERY, I., AND IACOVIDES, I. *Coping with water scarcity: Addressing the challenges*. Springer Science & Business Media, 2009.

[116] POLI, R. An analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications 2008* (2008), 1–10.

[117] POLI, R., KENNEDY, J., AND BLACKWELL, T. Particle swarm optimization. *Swarm Intelligence 1*, 1 (2007), 33–57.

[118] POSIK, P. Real-parameter optimization using the mutation step co-evolution. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on* (2005), vol. 1, IEEE, pp. 872–879.

[119] REYNOLDS, C. W. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph. 21*, 4 (1987), 25–34.

[120] RIEBSAME, W. E., CHANGNON JR, S. A., KARL, T. R., ET AL. *Drought and natural resources management in the United States. Impacts and implications of the 1987-89 drought*. Westview Press Inc., 1991.

[121] RITTER, A., AND MUNOZ-CARPENA, R. Performance evaluation of hydrological models: Statistical significance for reducing subjectivity in goodness-of-fit assessments. *Journal of hydrology 480* (2013), 33–45.

[122] RONKKONEN, J., KUKKONEN, S., AND PRICE, K. V. Real-parameter optimization with differential evolution. In *Proc. IEEE CEC* (2005), vol. 1, pp. 506–513.

[123] RUMELHART, D. E., MCCLELLAND, J. L., GROUP, P. R., ET AL. *Parallel distributed processing*, vol. 1. IEEE, 1988.

[124] SENTHIL BABU, S., AND VINAYAGAM, B. Surface roughness prediction model using adaptive particle swarm optimization (APSO) algorithm. *Journal of Intelligent and Fuzzy Systems* (2015), 345–360.

[125] SHI, Y., AND EBERHART, R. A modified particle swarm optimizer. In *IEEE International Conference on Evolutionary Computation Proceedings* (Anchorage, UK, 1998), pp. 69–73.

[126] SHI, Y., AND EBERHART, R. Parameter selection in particle swarm optimization. In *EP '98: Proceedings of the 7th International Conference on Evolutionary Programming VII* (London, UK, 1998), pp. 591–600.

[127] SHI, Y., AND EBERHART, R. Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington, DC, 1999), pp. 1945–1950.

[128] SIMON, D. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[129] SPENCER, H. *The principles of biology*. Williams and Norgate, 1864.

[130] STAGGE, J. H., TALLAKSEN, L. M., GUDMUNDSSON, L., VAN LOON, A. F., AND STAHL, K. Candidate distributions for climatological drought indices (SPI and SPEI). *International Journal of Climatology* (2015).

[131] SUGANTHAN, P. N., HANSEN, N., LIANG, J. J., DEB, K., CHEN, Y. P., AUGER, A., AND TIWARI, S. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Tech. rep., Nanyang Technological University, Singapore, 2005.

[132] TALLAKSEN, L., AND VAN LANEN, H. *Hydrological Drought: Processes and Estimation Methods for Streamflow and Groundwater*, vol. 48 of *Developments in water science*. Elsevier, 2004.

[133] TEAM, R. D. C. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[134] TRELEA, I. C. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters 85*, 6 (2003), 317–325.

[135] UY, N. Q., HOAI, N. X., MCKAY, R., AND TUAN, P. M. Initialising PSO with randomised low-discrepancy sequences: the comparative results. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* (2007), IEEE, pp. 1985–1992.

[136] VICENTE-SERRANO, S. M., BEGUERÍA, S., AND LÓPEZ-MORENO, J. I. A multiscalar drought index sensitive to global warming: the standardized precipitation evapotranspiration index. *Journal of Climate 23*, 7 (2010), 1696–1718.

[137] VRUGT, J. A., GUPTA, H. V., BASTIDAS, L. A., BOUTEN, W., AND SOROOSHIAN, S. Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resources Research 39*, 8 (2003), 1214.

[138] VRUGT, J. A., ROBINSON, B. A., AND HYMAN, J. M. Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces. *IEEE Transactions on evolutionary computation 13*, 2 (2009), 243–259.

[139] WANAS, N., AUDA, G., KAMEL, M., AND KARRAY, F. On the optimal number of hidden nodes in a neural network. In *Electrical and Computer Engineering, 1998. IEEE Canadian Conference on* (1998), vol. 2, pp. 918–921 vol.2.

[140] WANG, H., LI, C., LIU, Y., AND ZENG, S. A hybrid particle swarm algorithm with Cauchy mutation. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE* (2007), IEEE, pp. 356–360.

[141] WANG, W., VAN GELDER, P. H., VRIJLING, J., AND MA, J. Forecasting daily streamflow using hybrid ANN models. *Journal of Hydrology 324*, 1 (2006), 383–399.

[142] WEBER, M., NERI, F., AND TIRRONEN, V. Shuffle or update parallel differential evolution for large-scale optimization. *Soft Computing - A Fusion of Foundations, Methodologies and Applications 15* (2011), 2089–2107.

[143] WEISE, T. Global optimization algorithms-theory and application. *Self-Published,* (2009).

[144] WEISE, T., ZAPF, M., CHIONG, R., AND URBANEJA, A. J. Why Is Optimization Difficult? In *Nature-Inspired Algorithms for Optimisation*, R. Chiong, Ed., vol. 193 of *Studies in Computational Intelligence*. Springer, 2009, ch. 1, pp. 1–50.

[145] WILHITE, D. A., AND GLANTZ, M. H. Understanding: the drought phenomenon: the role of definitions. *Water international 10*, 3 (1985), 111–120.

[146] WOLPERT, D. H., AND MACREADY, W. G. No Free Lunch Theorems for Search. Tech. rep., Santa Fe Institute, 1995.

[147] WU, J., LONG, J., AND LIU, M. Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. *Neurocomputing 148* (2015), 136–142.

[148] WU, J., WANG, L., AND ZHU, B. The Meteorological Prediction Model Study of Neural Ensemble Based on PSO Algorithms. In *The Sixth World Congress on Intelligent Control and Automation* (Dalian, 2006), pp. 51–55.

[149] WU, X., AND ZHONG, M. Particle swarm optimization based on power mutation. In *Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on* (2009), vol. 4, IEEE, pp. 464–467.

[150] WYSS, G. D., AND JORGENSEN, K. H. A user's guide to LHS: Sandia's Latin hypercube sampling software. *Sandia National Laboratories* (1998).

[151] XIN, J., CHEN, G., AND HAI, Y. A Particle Swarm Optimizer with Multi-stage Linearly-Decreasing Inertia Weight. In *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization - Volume 01* (Washington, DC, USA, 2009), IEEE Computer Society, pp. 505–508.

[152] YAN, J., TIESONG, H., CHONGCHAO, H., XIANING, W., AND FALING, G. A shuffled complex evolution of particle swarm optimization algorithm. In *Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms, Part I* (Berlin, Heidelberg, 2007), ICANNGA '07, pp. 341–349.

[153] YANG, C.-H., HSIAO, C.-J., AND CHUANG, L.-Y. Linearly Decreasing Weight Particle Swarm Optimization with Accelerated Strategy for Data Clustering. *IAENG International Journal of Computer Science 37*, 3 (2010).

[154] ZAMBRANO-BIGIARINI, M., AND ROJAS, R. A model-independent Particle Swarm Optimisation software for model calibration. *Environmental Modelling and Software* (2013).

[155] ZHANG, B.-L., AND DONG, Z.-Y. An adaptive neural-wavelet model for short term load forecasting. *Electric Power Systems Research 59*, 2 (2001), 121–129.

[156] ZHANG, J., AND DING, X. A multi-swarm self-adaptive and cooperative particle swarm optimization. *Engineering applications of artificial intelligence 24*, 6 (2011), 958–967.

[157] ZHANG, X., SRINIVASAN, R., ZHAO, K., AND LIEW, M. V. Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. *Hydrological Processes 23*, 3 (2009), 430–441.

[158] ZHOU, C., GAO, L., GAO, H., AND PENG, C. Pattern classification and prediction of water quality by neural network with particle swarm optimization. In *The Sixth World Congress on Intelligent Control and Automation* (Dalian, 2006), pp. 2864 – 2868.