Water Resour Manage (2017) 31:3205–3233
DOI 10.1007/s11269-017-1694-6

Modern Optimization Methods in Water Resources Planning, Engineering and Management

Gokmen Tayfur[1]

Received: 19 October 2016 / Accepted: 27 April 2017 /
Published online: 16 May 2017

Abstract Mathematical (analytical, numerical and optimization) models are employed in many disciplines including the water resources planning, engineering and management. These models can vary from a simple black-box model to a sophisticated distributed physics-based model. Recently, development and employment of modern optimization methods (MOMs) have become popular in the area of mathematical modeling. This paper overviews the MOMs based on the evolutionary search which were developed over mostly the last 30 years. These methods have wide application in practice from finance to engineering and this paper focuses mostly on the applications in the area of water resources planning, engineering and management. Although there are numerous optimization algorithms, the paper outlines the ones that have been widely employed especially in the last three decades; such as the Genetic Algorithm (GA), Ant Colony (AC), Differential Evolution (DE), Particle Swarm (PS), Harmony Search (HS), Genetic Programming (GP), and Gene Expression Programming (GEP). The paper briefly introduces theoretical background of each algorithm and its applications and discusses the merits and, if any, shortcomings. The wide spectrum of applications include, but not limited to, flood control and mitigation, reservoir operation, irrigation, flood routing, river training, flow velocity, rainfall-runoff processes, sediment transport, groundwater management, water quality, hydropower, dispersion, and aquifers.

Keywords Genetic algorithm · Ant colony · Differential evolution · Particle swarm · Harmony search · Genetic programming · Gene expression programming · Water resources

1 Introduction

Models can be used in planning, design, and operation of water resources projects. They can be employed for many purposes, such as planning and designing soil

✉ Gokmen Tayfur
gokmentayfur@iyte.edu.tr

[1] Department of Civil Engineering, Izmir Institute of Technology, Urla, 35430 İzmir, Turkey

conservation practices, irrigation water management, wetland restoration, stream restoration, water-table management, flood protection, rehabilitation of aging dams, reservoir operation, floodplain management, water-quality evaluation, and water-supply forecasting, among many. They can also be employed to understand dynamic interactions between climate and land-surface hydrology, and to quantify the impacts of management strategies, linking human activities within the watershed to water quantity and quality of the receiving streams or reservoirs for environmental and water resources protection (Singh and Woolhiser 2002).

Models in water resources engineering vary from a simple black-box model to a distributed physics-based model. Black box models relate input variables to output variable without concentrating on a detailed description of the physical processes. The use of time-series methods is complicated by non-stationarity and non-linearity in the data that require, in turn, experience and expertise from a modeler (Imrie et al. 2000). The physics-based models accounting for physical processes can provide more accurate results at the expense of overwhelming data requirements, simplifications in the governing equations, and sophisticated numerical techniques (Singh and Woolhiser 2002).

One of the recent developments in hydrologic, hydraulic, and water resources engineering modeling is the employment of the modern optimization methods (MOMs) based on the evolutionary search, such as such as Genetic Algorithm (GA), Ant Colony (AC), Differential Evolution (DE), Particle Swarm (PS), Harmony Search (HS), Genetic Programming (GP), and Gene Expression Programming (GEP). The developments of these methods started in the late seventies and the first applications to hydrological, hydraulic, and water resources engineering problems happened to be after 1990s. These techniques are effective at handling dynamic, nonlinear, and noisy data, especially when the underlying relationships are not fully understood (See and Openshaw 2000). The reason for an increasing interest in these models has resided in their intrinsic generality, flexibility, and global performance in most applications where other models either tend to fail or become cumbersome.

Optimization can be described, in simple terms, as the maximization or minimization real function by systematically choosing input value of the function. It can be more generally expressed as the finding of the best available values of some objective function given a defined domain under a set of constraints. The use of optimization methods in water resources planning and management starts in early 1960s with the use of linear programming in which the objective function is linear and the constraints are specified using only linear equalities and inequalities. Kantorovich (1939) introduced the linear programming theory and the simplex solution algorithm was developed by Dantzig (1953) who wrote the first book in 1963 on the linear programming. Many application of the linear programming in solving water resources engineering optimization problems can be found in Loucks et al. (1981). Since then, the method has also been used for optimal pump scheduling in water supply networks (Jowitte and Germanopoulos 1992), use of different quality irrigation waters (Afzal et al. 1992), water supply headworks modeling (Kuczera 1993), reservoir operations (Tao and Lennox 1991), groundwater management (Lall and Lin 1991), optimal researvoir opeartions (Crawley and Dandy 1993), flood control (Needham et al. 2000), optimal pump scheduling (Demotier et al. 2001; Pasha and Lansey 2009), and among many.

The dynamic programming (DP) is, on the other hand, a method for solving a complex problem by breaking it down into a collection of simpler subproblems. Then, it solves each of those subproblems just once, and stores their solutions (Bellman 1957). Next time the same subproblem occurs, one simply looks up the previously computed solution instead of recomputing its solution. In short, the dynamic programming algorithm examines the previously solved subproblems and combines their solutions to give the best solution for the given problem (Nemhauser 1966). Unlike the linear programming, the problem can be linear and nonlinear in the case of dynamic programming (Loucks et al. 1981). Hall and Howell (1963) are the first ones who applied DP to optimize a reservoir design. McKerchar (1975) used DP for the operation of hydroelectric power storage. Kolo and Haimes (1977) employed DP for the capacity expansion of regional water resources system. Guitron (1981) used DP for the optimal operation of multipurpose reservoir. Many application of dynamic programming in solving water resources engineering problems can be found in Loucks et al. (1981). Since then, the method has also been used for the reservoir operation (Marino and Mohammadi 1983), water supply system (Kareliotis 1984), multireservoir modeling (Chandramouli and Raman 2001), hydropower reservoir operation (Zhao et al. 2014), among many. Singh (2012) summarizes LP and DP algorithms and gives some applications in an overview paper.

Starting with Fermant and Newton in seventeenth century and Lagrange and Gauss in eighteenth century, calculus-based formulas for identifying optima were developed analytically and iteratively throughout the centuries. Until the advent of heuristic (modern) methods, all these algorithms required the differentiability and thus continuity of the objective function. With the advances in computing capabilities, in the last four decades, heuristic optimization methods have been developed. These are basically nonlinear search and optimization methods that do not rely on mathematical properties of a function, such as the continuity and the differentiability. They search the solution space randomly and they are population based and data driven. There are numerous MOMs, yet this paper briefly introduces the commonly employed evolutionary optimization ones in water resources engineering with their applications, such as GA, AC, DE, PSO, HS, GP, and GEP.

2 Genetic Algorithm (GA)

The genetic algorithm (GA) was first introduced by Holland (1975) who was inspired by the biological processes of natural selection (i.e. Darwinian evolutionary view), and the genetic operations (i.e. Mendelian genetics). The first engineering application of GA by Goldberg (1983) triggered the use of it in different disciplines, including the water resources engineering.

2.1 GA Basics

GA has four basic units of *bit*, *gene*, *chromosome*, and *gene pool*. *Bit* is the basic building block which is represented by a digit of 1 or 0. The combination of bits forms a *gene* representing a model parameter. The attachment of genes forms a *chromosome* standing for a possible solution. *Gene pool* is formed by many individual chromosomes (Tayfur 2012).

GA algorithm have four basic operations of *generation of initial gene pool*, *calculation of fitness for each chromosome*, *selection of chromosomes*, *operation of cross-overing* and *mutation* (Tayfur 2012). Uniform distribution or a normal distribution can be employed to randomly generate initial chromosomes for the gene pool (Sen 2004). Fitness of each chromosome can be evaluated by first substituting each chromosome into objective function to find their values *f(C$_i$)* and then obtaining their fitness by Eq. 1 (Sen 2004) as follows:

$$F(C_i) = \frac{f(C_i)}{\sum_{i=1}^{N} f(C_i)} \tag{1}$$

where, N is number of chromosomes in the gene pool, C_i is chromosome i, $f(C_i)$ is value of objective function for chromosome i, and $F(C_i)$ is fitness value for chromosome i.

Selection of chromosomes after the evaluation of their fitness values can be performed randomly. There are methods available for the selection process such as roulette wheel and ranking (Sen 2004). After the selection process, pairs (parent chromosomes) are first formed and they are then subjected to the cross-over operation by interchanging the genes. The last operation in a single iteration is the mutation by which bits are reversed (i.e., *1 to 0* or *0 to 1*).

By these operations, it intends to search the solution space thoroughly. Fig. 1 shows an example for crossover and mutation operations. As seen, first two chromosomes (parent chromosome I and parent chromosome II) are subjected to the cross-over by the single cut from the third digit on the left, yielding new chromosomes (off-spring I and off-spring II) at the bottom. The values of 187 and 106 become 123 and 170, respectively after the crossover operation. By the mutation operation, fourth digits from the left of the off-springs are reversed. Thus, the final version of the off-springs becomes 107 and 186, respectively at the bottom. By the basic GA operations, large portion of solution space is searched to reach the global optimum. Fig. 2 shows the flowchart on how the GA algorithm works. One can find more details on GA elsewhere (Goldberg 1999; Sen 2004; Tayfur 2012).

2.2 GA Applications

First applications of GA in the area of water resources engineering starts in early 1990s with the work of Wang (1991) who calibrated rainfall-runoff conceptual model prameters. McKinney and Lin (1994) effectively and efficiently employed GA in groundwater simulation model to solve maximum pumping rate, minimum cost water supply development, and

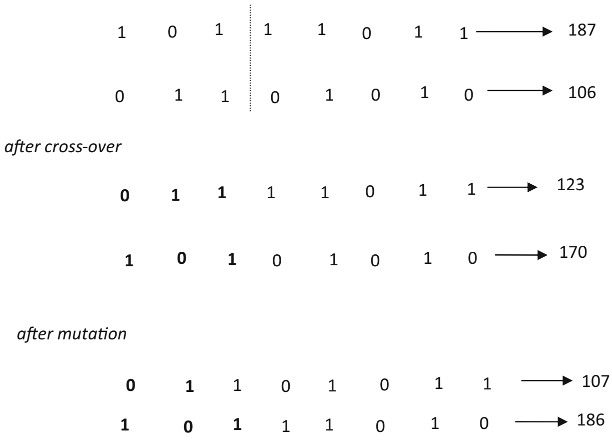


Fig. 1 Example for crossover and mutation operations

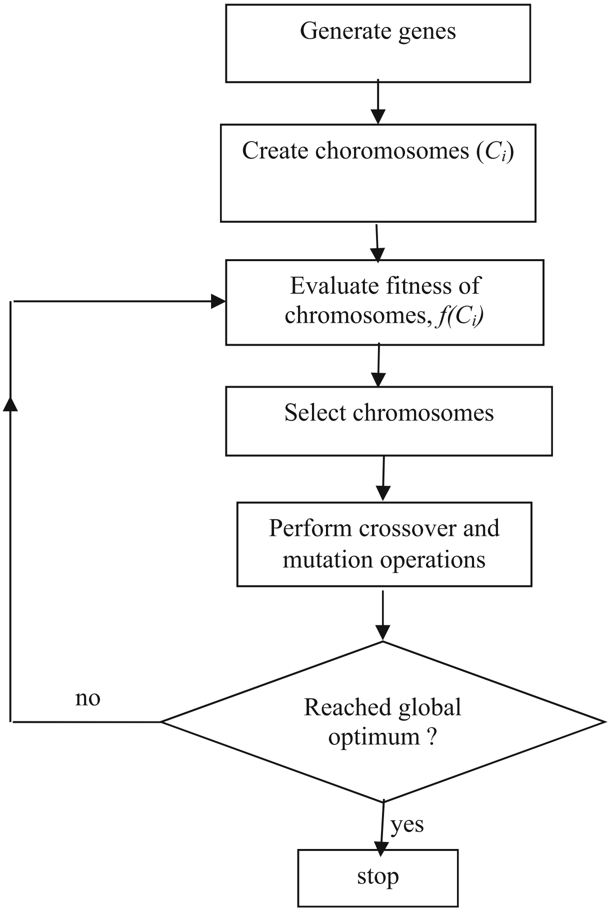

Generate genes
Create choromosomes (C_i)
Evaluate fitness of chromosomes, $f(C_i)$
Select chromosomes
Perform crossover and mutation operations
Reached global optimum ?
no
yes
stop

Fig. 2 Flow chart for GA algorithm

minimum cost aquifer remediation. Chang and Chen (1998) used real coded and binary coded GA to select good parameters for the operation of a simplified flood control reservoir.

Jha et al. (2004) sucessfully used GA to obtain hydraulic characteristic parameters of well loss coefficient, and aquifer loss coefficient from step-drawdown test data of production wells. They pointed out that proper selection of population size is essential to ensure the efficient optimization. Raju and Kumar (2004) employed GA to obtain efficient cropping pattern for maximizing benefits for an irrigation project in India. In their study, the constraints included the continuity equation, land and water requirements,crop diversification and restrictions on the storage. They employed the penalty function approach to convert constrained problem into an unconstrained one. For fixing GA parameters, they ran the model for various values of population, generations, cross over and mutation probabilities.

Jothiprakash and Shanthi (2006) applied GA to Pechiparai reservoir in Tamil Nadu, India to derive the optimal operational strategies. They set the objective function to minimize the annual sum of squared deviation form desired irrigation release and desired storage volume. The decision variables were the release rate for irrigation and other demands from the reservoir. Karahan et al. (2008) employed GA to obtain optimal values of the coefficents of intensity-duration-frequency relations using measured rainfall data from several stations in

Turkey. Tayfur and Moramarco (2008) predicted event based real hydrographs, based on the rating curve formulations using stage data only. In a similar fashion, Tayfur et al. (2009) performed hydrograph predicitions using the rating curve formulation employing discharge data. In both studies, optimal values of the parameters of the formulations were obtained by the GA using several event hydrographs in the calibration phase. They successfully predicted different hydrogrpahs observed on different dates in different stations of Tiber River in centaral Italy in the verification phase. They showed that GA has no extrapolation capability problem, unlike the artificial neural networks (ANN), in prediciting hydrographs. Tayfur (2009) employed 80 sets of field data of flow and channel characteristics, together with dispersion coefficent from 30 rivers in the USA to perform dispersion coefficient predictions using several empirical formulations whose parameters were optimized using the GA.

Jothiprakash et al. (2011) developed operational policy for a multi-reservoir system in India using the GA and concluded that the robostic, probabilistic, random search method (GA) gives better optimal operating policies for a multi-reservoir system than the stochastic dynamic programming (SDP). Their study also showed that the increase in number of variables in multi-reservoir system could be handled by increasing the probability of crossover and increasing the number of generation in GA model. Tayfur and Singh (2011) predicted bankfull and mean discharge using empirical models based upon channel cross-section by expert methods including GA. Sahay (2012) predicted transient storage model parameters of rivers using 58 sets of data from 33 different rivers. The data included flow, channel and dispersion observations. He/she proposed several empirical models whose parameters were predicted by GA. He/she calibrated and verified the performance of GA_based empirical model. Tayfur et al. (2013) and Tayfur and Karimi (2014) investigated sediment load predictions and generalization from laboratory scale to field scale using the principle component analysis (PCA) in conjunction with GA. Five main dimensionless parameters for total load are identified by using PCA. These parameters are then used in constructing nonlinear equations whose coefficeints are obtained by GA. Tayfur and Karimi (2014) also proposed a new equation based on the Bagnold approach. The optimal values of the coefficients of this equation were also obtained by GA. The performance of the proposed equation was found to be very efficient. Kalita et al. (2014) evaluated the appropriate river training measures in a particular reach of a river. They linked the optimization model GA with 2D hydrodynamic model for obtaining cost effective combination of groynes which could maintain a user defined flow speed in a pre-decided portion of a river reach. They obtained reasonable results and pointed out that the computational time required for such GA_based linked simulation-optimization model is high, in agreement with Ayvaz and Karahan (2008) and Ayvaz (2016) who identified unknown groundwater well and pollution source locations, respectively.

Zucco et al. (2015) proposed a procedure for reverse flood routing in natural channels for three different configurations of hydrometric monitoring of a river reach where lateral flow is significant and no rainfall data are available for the intermediate basin. The first configuration considers only the downstream channel end as a gauged site where discharge and stages are recorded. The second configuration assumes the downstream end as a gauged site but only in terms of stage. The third configuration envisages both channel ends equipped to recording stages. The parameters, coefficients and exponents of the models are obtained, for each configuration, using the genetic algorithm method. They were successfully able to generate upstream flow hydrographs measured at different stations on Tiber River. This was a very crucial outcome for prediciting hydrographs at the ungauged stations. Chiang and Willems (2015) combined GA with the Model Predictive Control (MPC) technique to develop and test a real-time flood control method for the 12 gated weirs in the Belgian case study of the river Demer. The combination of the models was able to cope with the highly nonlinear system behaviour and local minimum problems. Reshma et al. (2015) optimized the calibration of event based watershed model parameters using GA. Perea et al. (2016) proposed a new approach for sustainable management of pressurized irrigation networks using multiobjective genetic algorithms. Their model established the optimal sectoring operation during the irrigation season that maximized farmer's profit and minimized energy cost at the pumping station whilst satisfying water demand of crops at hydrant level taking into account the soil water balance at farm scale. Their methodology was applied to a real irrigation network in Southern Spain.

3 Ant Colony (AC)

Ant colony (AC) was first proposed by Dorigo (1992) who was inspired by the ants that are completly blind but could manage to establish the shortest path between thier nest and food source. We now know that ants communicate information by depositing a pheromone trail. A moving ant leaves some pheromone on the ground to mark its way. An isolated ant can detect previously laid pheromone trail and follow it, thus reinforcing the track with its own pheromone. Hence, it is highly probable that a path which was previously chosen by many ants would be chosen by the new ants.

Fig. 3 is a conceptual representation of AC where there are two paths from ant nest (Location A) to food source (Location B). Between A and B locations, there is an obstacle. AC = CD = DE = EB = 1 unit and CF = FE = 2 unit. So, the short path (ACDEB) is 4 units and the long path (ACFEB) is 6 units. Let us assume that there are 16 ants in the nest and they are about to search for food. Each ant moves one unit distance per unit time. Each ant deposits one unit pheromone (p) on the path after reaching the next node. At initial stage (t = 0), according to Figs. 4, 16 ants move in AC direction. At 1 unit time (t = 1), 16 ants arrive at C node,

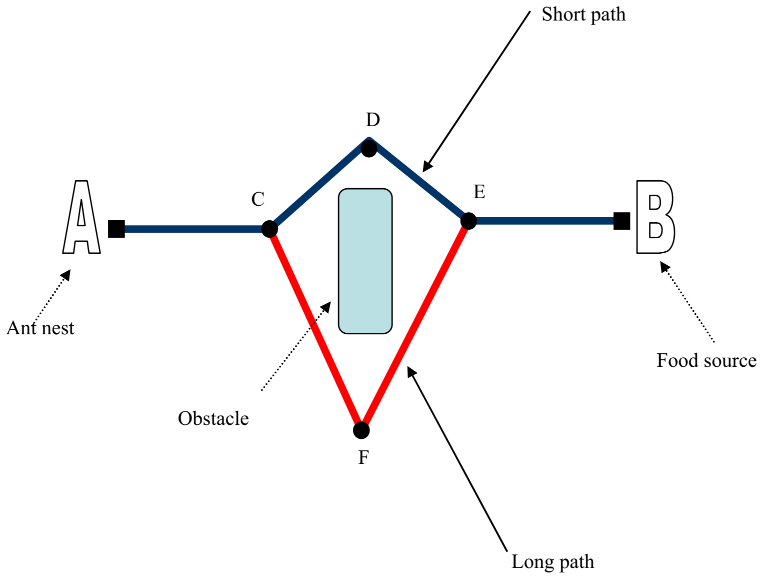


Fig. 3 Conceptual representation of ant colony behaviour

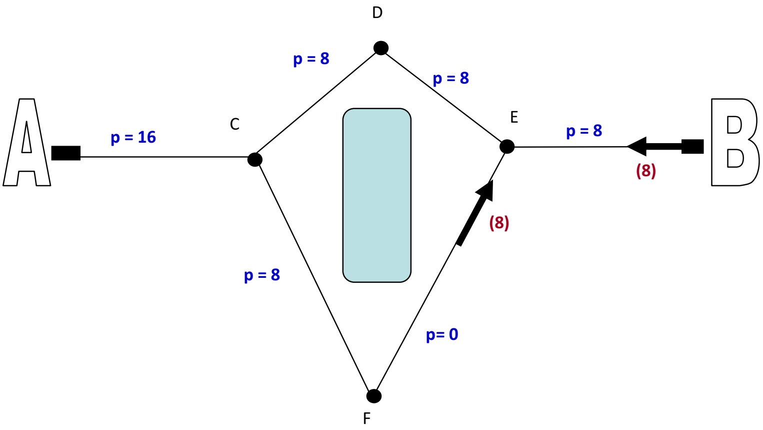


Fig. 4 Pheremone concentrations and ant locations at the end of 4 unit time

leaving 16 units p on AC. At node C, to be fair, we can assume that 8 ants choose CD and 8 choose CF paths. At the end of 4 units time (*t* = 4), 8 ants on the short track arrive at food source B, while other 8 is just half way of FE, as shown in Fig. 4. At *t* = 5, when 8 ants on short route arrive back at node E, there is already 8 units of p on ED and 8 units of p on EF. Therefore, it is assumed that, with equal probability, 4 ants would choose ED and other 4 would chose EF.

As ants continue their journey back and fourth, as presented in Fig. 5, at 13 units time (*t* = 13); 3 would be at node E from the food source, 5 on the short route would be at E for the food source, 1 on the long route would be at E for food source, 3 would be at F, 4 ants from the nest would be at node C. When 4 ants from the nest reach C; there is already 24 units of p on CD and 20 on CF. With high probability more ants would choose CD. Hence, it is assummed that 3 would chose CD and 1 CF. When 3 ants from the food source reach node E, there is already 24 p on DE and 17 on FE. Therefore, it is assumed that 2 would choose ED and 1 EF. As seen in Fig. 5, the pheromone concentration on the short route is already higher and

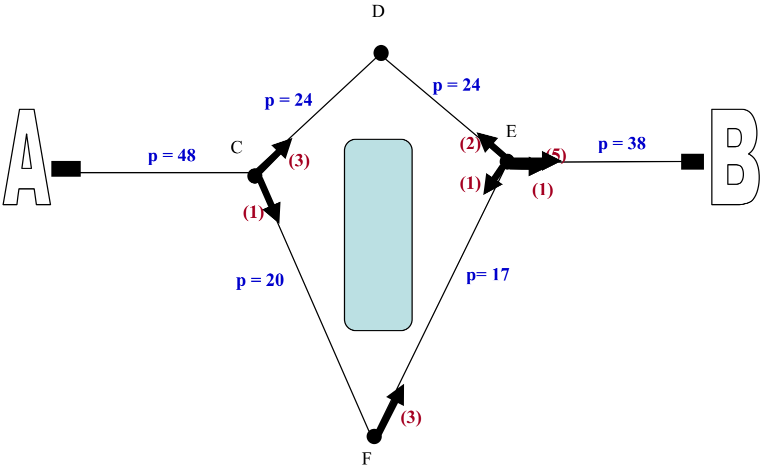


Fig. 5 Pheremone concentrations and ant locations at the end of 13 unit time

therefore as time progresses, more ants with high probability would choose the short path. Eventually all the ants would be on the short path.

3.1 AC Algorithm Basics

The algorithm is underlined by the basic principle of positive reinforcement via the use of pheromone trails. It adopts a stochastic decision making policy using local information. It makes use of a colony of cooperating individuals. It can incorporate features such as memory and sight that are not found in natural ants (Dorigo et al. 1996). Pheromone updates can only occur when one or more ants have completed their tour.

Fig. 6 shows the flowchart for AC algorithm where the main steps can be summarized as follows:

1. Trial solutions are incrementally constructed as artificial ants move from one decision point to the next until all decision points have been covered.

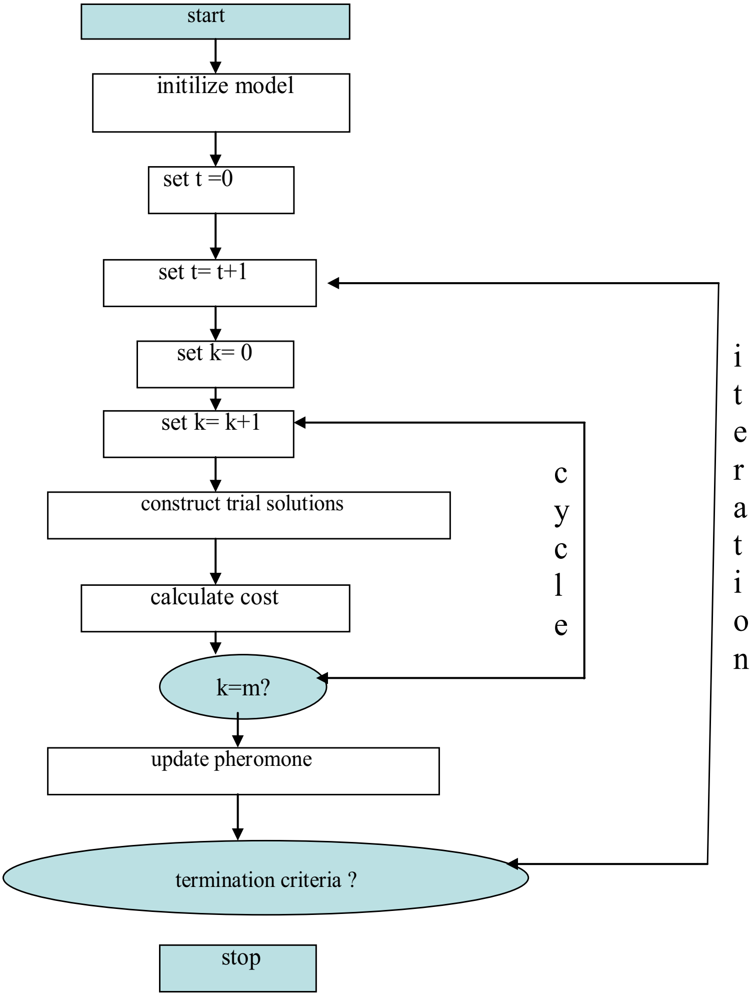


Fig. 6 Flowchart for AC algorithm

2. The cost of the generated trial solution is calculated. The generation of a complete trial solution and calculation of the corresponding cost is referred to as a cycle (k). One cycle is the completion of path from nest to source by one ant and the related cost.
3. Pheromone is updated after the completion of one iteration (t), which consists of m cycles, i.e., m is the number of used ants. One iteration stands for the case when all the ants cover the path.

The component to be added to the trial solution at each decision node is expressed as (Dorigo et al. 1996):

$$P_{i(j)}(k,t) = \frac{\left[p_{i(j)}(t)\right]^{\alpha}\left[\eta_{i(j)}(t)\right]^{\beta}}{\sum_{L_{i(j)}}\left[p_{i(j)}(t)\right]^{\alpha}\left[\eta_{i(j)}(t)\right]^{\beta}} \tag{2}$$

where; $P_{i(j)}$ (k,t) is the probability that option $L_{i(j)}$ is chosen at cycle k and iteration t; $p_{i(j)}(t)$ is concentration of pheromone associated with option $L_{i(j)}$ at iteration t; $\eta_{i(j)} = L/c_{i(j)}$ is the heuristic factor favoring options that have smaller local costs; $c_{i(j)})$ is the set of costs associated with options $L = (l_{i(j)})$ and α and β are the exponent parameters that control relative importance of pheromone and local heuristic factor, respectively.

The probability depends on existing pheromone concentrations deposited on the paths. The vision of the ants, in turn, depends on the cost of the paths. It should be noted that the addition of Local heuristic Factor $\eta_{\ i(j)}$is Analogous to providing real ants with sight, and it is sometimes called *visibility.* Artificial ants can also be provided with memory to ensure that each decison point is only visited once.

Once an iteration is completed, and *m* trial solutions are constructed, the pheromone trails are updated in a way to reinforce good solutions. The general form of the pheromone updating equation is given as follows (Dorigo et al. 1996):

$$p_{i(j)}(t+1) = \delta p_{i(j)}(t) + \Delta p_{i(j)} \tag{3}$$

where; $p_{i(j)}$ *(t + 1)* is pheromone concentration associated with option $L_{i(j)}$ at iteration (t + 1) and Δ $p_{i(j)}$ is the change in pheromone concentration associated with option $L_{i(j)}$ as a function of the trial solutions found at iteration *t.* δ is the pheromone persistance coefficient and it takes values less than 1.

δ enables greater exploration of the search space, avoids premeture convergance to suboptimal solutions, and reduces the likelihood that high cost solutions would be selected in future cycles. The details of AC algorithm can be obtained from Dorigo (1992), among many.

3.2 AC Algorithm Applications

The first applications of AC in water resources engineering area start in early 2000s. Abbaspour et al. (2001) employed AC for estimating unsaturated soil hydraulic parameters. Maier et al. (2003) applied AC algorithm to find optimal design of water distribution systems. Jalali et al. (2005) proposed an improved version of the AC algorithm in single water-reservoir operation optimization. Li et al. (2006) used AC for parameter estimation in groundwater. Kumar and Reddy (2006) employed AC to derive operating policies for a multi-purpose

Hirakud reservoir system in India. They approached the problem by considering a finite time series of inflows, classifying the reservoir volume into several class intervals, and determining the reservoir release for each short and/or long period with respect to a predefined optimality criterion. The multiple objectives were comprised of minimizing flood risks, minimizing irrigation deficits and maximizing hydropower production. Ibanez et al. (2008) used AC for the optimal control of pumps in water distribution networks.

Tu et al. (2011) employed AC for designing optimal irrigation system. The objective function was the minimization of the specific energy consumption subject to the constraints of pipe diameters, number of sprinklers and working pressure of the end sprinkler along the pipeline and pump-pipeline cooperation conditions. Ostfeld (2011) presented a review paper on AC for water resources systems analysis. Most recently, Massoumi et al. (2015) also published a review paper in which one can find AC applications on reservoir operation, surface water management, water distribution system, groundwater management, environmental management, watershed management, etc. Afshar et al. (2015) point out that AC requires expensive computations which can limit their suitability for implicit and/or explicit stochastic optimization. They also state that mathematical analysis of the algorithm remains implicit despite its its popularity in solving nonlinear, nonconvex, multimodal problems for which deterministic search techniques incur difficulty or fail completely.

4 Differential Evolution (DE)

Differential evolution (DE) was first introduced by Storn (1996) in a conference paper and by Storn and Price (1997) in a journal paper. DE is an evolutionary method which does not require the optimization function to be continuous and thus differentiable. It is a population based and it uses, like GA, similar operations such as selection, crossover, and mutation. It relies more on mutation and optimizes a problem by maintaining a population of candidate solutions, creating new candidate solutions, and then keeping the ones having the best fitness (Storn and Price 1997).

4.1 DE Algorithm Basics

In DE, the population of solution vectors (NP) is randomly created at the beginning and successfully improved by applying mutation, crossover and selection operators (Karaboga and Okdem 2004). Normally, *NP* should be about 10 times the number of parameters in a vector (Storn and Price 1997; Karahan 2011). Non-uniform crossover is used in DE, by which offspring vector parameters are taken more often from one parent.

DE maintains two arrays, of which first one holds the current vector population, while the second array accumulates vectors that are selected for the next generation. When X_a and X_b are chosen randomly, their weighted differential is used to perturb another randomly chosen vector X_c. This process can be mathematically expressed as (Vasan and Raju 2007):

$$X_c^{'} = X_c + F(X_a - X_b) \tag{4}$$

where, F is a user-specified weighting factor whose optimal value is suggested to be 0.5 to 1 (Price and Storn 2005). In every generation, each primary array vector of X_i is targeted for crossover with a vector of X'_c to produce a trial vector of X_t. Thus, the trial vector becomes the offspring of two parents, i.e., (1) a noisy random vector and (2) target vector against which trial vector must compete. Nonuniform crossover is used with a crossover constant (CR) whose optimal range is 0.5 to 1 (Price and Storn 2005). CR represents probability that child vector inherits parameter values from noisy random vector. The cost of the trial vector is compared to that of the target vector. The vector having the lowest cost of the two would survive for the next generation. This process is continued until termination criterion is satisfied. Fig. 7 presents flowchart for DE algorithm.

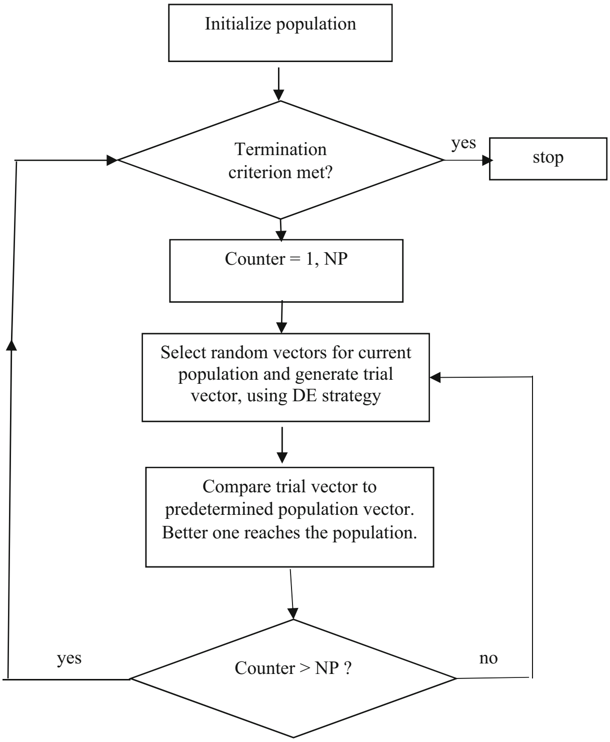
Population size (NP), weighting factor (F) and crossover rate (CR) are the major parameters controlling the evolution in DE algorithm. Price and Storn (2005) gave 10 different strategies that can be adopted in DE. The strategies depend on the problem type and they may vary based on perturbed vector, number of difference vectors used in perturbation and the employed crossover type.

DE can handle functions that are nondifferentiable, nonlinear, multimodal having complex search space with many local optimal solutions (Price and Storn 2005; Vasan 2005). The details of DE can be obtained from the literature (Storn and Price 1997; Karaboga and Okdem 2004; Price et al. 2005).

4.2 DE Algorithm Applications

We see first application of DE in water resources engineering area by Vasan and Raju (2004) who applied it to a case study of Bisalpur project, Rajasthan, India to determine suitable

Fig. 7 Flowchart for DE algorithm


Initialize population
Termination criterion met?
yes
stop
Counter = 1, NP
Select random vectors for current population and generate trial vector, using DE strategy
Compare trial vector to predetermined population vector. Better one reaches the population.
yes
Counter > NP ?
no

cropping pattern that would yield maximum net benefits. Then, Vasan and Raju (2007) applied DE to a case study of Mahi Bajaj Sagar Project (MBSP), India for efficient irrigation planning. They employed 10 different strategies of DE with extensive sensitivity analysis performing 3600 combinations of population size, crossover constant and weighting factor. Their study revealed that careful selection of strategy and parameters is essential for a meaningful analysis of the problem. This requires extensive sensitivity analysis and assessment of the robustness of the methodology (Vasan and Raju 2007).

Li et al. (2013) applied DE for the prediciton of longitudinal dispersion coefficents in natural rivers. They proposed a new expression considering hydraulic and geometric characteristics of rivers. They used the mutation operator, crossover operator, selection operator to generate the offspring. They employed 65 data sets from 29 natural streams. They concluded that DE has the ability to find the least root mean square error, the highest coefficient of correlation, the least average relative error to the measure value, the highest accuracy, better convergence characteristics, and computational efficiency. Jothiprakash and Arunkumar (2013) applied DE for maximizing the hydropower production from a reservoir.

Gurarslan and Karahan (2015) employed DE for solving inverse problem of groundwater pollution source identification. Recently, Jia et al. (2016) implemented DE for obtaining best compromise solution of real time flood mitigation operations of multi reservoir system. The multi-objective flood operation problem in Shiguan River Basin (in China) contained two reservoirs and three flood control points. The analyses were performed to compare four historical flood operations scenarios, their model and current operating rules (Jia et al. 2016). Although DE demonstrated its effectiveness for flood operation in multireservoir systems, they raised points on its flexibility, convergance ability, and computational efficiency when the system dimensions are further increased, such as an increase number of reservoirs and an increase number of flood control points.

5 Particle Swarm (PS)

Particle swarm (PS) method was first introduced by Kennedy and Eberhart (1995) in IEEE Conference. It is inspired by the movement of bird (or fish) flock. It is an evolutionary based computational method optimizing a problem iteratively. It solves the problem by having a population (particles) of candidate solutions. It moves these particles around in the search-space according to simple mathematical formulae on the particle's position and velocity. Each particle's movement is influenced by its local best known position, which is guided toward the best known positions in the search-space. Each particle adjusts its flying according to the experiences of both itself and its companions. The best positions updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. During the process; overall best value attained by all the partciles within the group and the coordinates of each element in hyperspace associated with its previous best fitness solution are recorded in the memory (Chau 2007; Kumar and Reddy 2007). The important advantages of PS algorithm are its relatively simple coding, low computational cost, fast converges, and adaptiveness corresponding to the change of the best group value (Chau 2007).

Although PS algorithm is similar to GA with respect to fitness concept and random population initialization, the evolution of generations in such a system occurs by cooperation and competition. The population responds to the quality factors of the previous best individual and group values (Kumar and Reddy 2007).

PS is a metaheuristic algorithm which makes few or no assumptions about the problem being optimized and it can search very large spaces of candidate solutions. However, as it is the case for the other evolutionary population based algorithms, it does not guarantee an optimal solution. PSO does not use any evolution operators, such as selection, crossover and mutation. In PSO, particles fly through the problem space by following its own experience and the best experience attained by the swarm as a whole (Shourian et al. 2008).

5.1 PS Algorithm Basics

PS algorithm consists of number of particles moving around in the search space. Each particle represents a possible solution to the problem at hand. In each iteration, the velocity of each particle is updated at each iteration as follows (Shourian et al. 2008):

$$v_{i+1} = \lambda\left(wv_i + c_1r_1(p_i - x_i) + c_2r_2\left(p_g - x_i\right)\right) \tag{5}$$

where x_i is particle's position vector, v_i is particle's velocity vector, p_i is position of the best candidate solution encountered by the particle, p_g is the overall (global) best found position, attracting the other PS particles, λ is constriction coefficient which can be taken as unity (Ostadrahimi et al. 2012), w is inertia weight representing the exploration and exploitation properties of algorithm, taking values in (0.4–0.9) (Moghaddam et al. 2016), c_1 and c_2 are acceleration coefficients, taking usually value of 2 (Afshar et al. 2013), and r_1 and r_2 are random values in between (0, 1) (Afshar et al. 2013).

Each particle's position is updated at each iteration by adding the velocity vector to the position vector as (Shourian et al. 2008):

$$x_{i+1} = x_i + v_{i+1} \tag{6}$$

The particles have no neighborhood restrictions, i.e., each particle can affect all other particles. PS algorithm updates the inertia weight at each iteration as (Shourian et al. 2008):

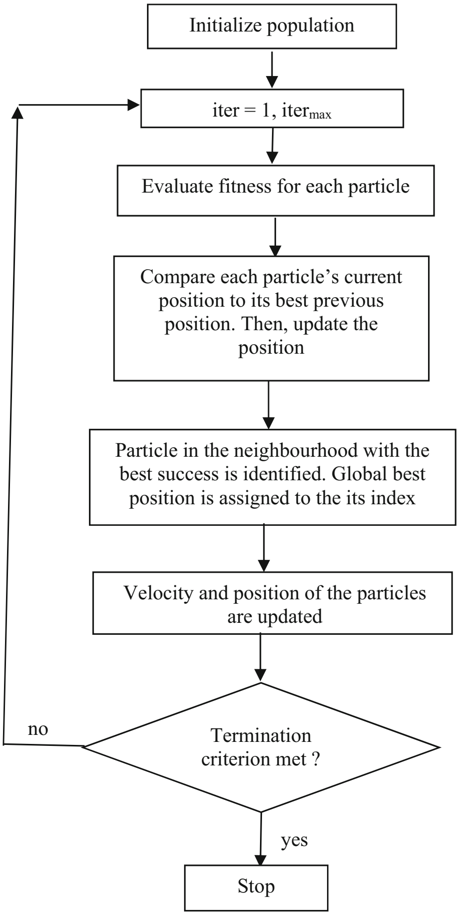
$$w_{\mathrm{iter}} = w_{\max} - \frac{w_{\max} - w_{\min}}{\mathrm{iter}_{\max}}.\mathrm{iter} \tag{7}$$

where w_{iter} is inertia weight in each iteration, $\mathrm{iter}_{\max}$ is maximum number of iterations, and $w_{\max}$ and $w_{\min}$ are maximum and minimum inertia weights, respectively. PS starts with a set of randomly generated solutions and then updates the swarm using Eqs. 5 and 6 at each iteration. This process is continued until the stopping criteria are satisfied. Fig.8 shows the flowchart for PS algorithm. The details of PSO is given elsewhere (Shi and Eberhart 1998; Kennedy and Eberhart 2001; Chau 2007; Gurarslan and Karahan 2011; Karahan 2012).

5.2 PS Algorithm Applications

We see first applications of PS in the area of water resources engineering in the second half of 2000s, with studies of Chau (2007) and Kumar and Reddy (2007). Chau (2007) employed PS for river stage forecasting while Kumar and Reddy (2007) used it for multipurpose reservoir operation. Shourian et al. (2008) employed PS together with flow simulation model MODSIM

Fig. 8 Flowchart for PS algorithm


Initialize population
iter = 1, iter$_{max}$
Evaluate fitness for each particle
Compare each particle's current position to its best previous position. Then, update the position
Particle in the neighbourhood with the best success is identified. Global best position is assigned to the its index
Velocity and position of the particles are updated
Termination criterion met ?
no
yes
Stop

to size water storage, transfer components of a river basin, and allocate water resources optimally over time and space among competing demands, considering coordinated operation of the system components. The objective function was the maximization of the total net benefit consisting of benefits due to water supply to different types of users, construction costs of dams, water transfer, and pumping system.

Afshar et al. (2011) used PS to automatically calibrate the parameters of water quality model of CE-QUAL-W2. The approach was sucessfully applied in temperature and water budget calibration of Karkheh reservoir in Iran. The use of PS overcame problem of high computational efforts required when a conventional calibration search technique was used. Ostadrahimi et al. (2012) used multi-swarm PS in conjuction with HEC-ResPRM simulation model for the purpose of developing operation rules for multi-reservoir system. Karahan (2012), using PS, determined optimal values of the parameters of the rainfall-intensity-duration-frequency relations. Gaur et al. (2013) used PS together with ANN for the managment of groundwater resources of Dore river basin in France. The Analytic Element Method (AEM) based flow model was used to generate data. The ANN-PSO model was successfully applied to minimize the pumping cost of the wells, and the pipe line. The discharge and location of the pumping wells were taken as the decision variable and theANN-PSO model efficiently found the optimal location of the wells.

Moghaddam et al. (2016) estimated optimal values of the parameters of the Muskingum model. PS performed high accuracy along with a fast rate of convergence. When its parameters are set right, PS can achieve the nearly global optimum solution in few iterations. Moghaddam et al. (2016) revealed that the most sensitive parameters in PS algorithm were c_1 and c_2 whose best values were found to be $c_1 = c_2 = 2.05$. Most recently, Spiliotis et al. (2016) adopted PS to achieve the identification of optimal hedging rules for operating reservoir systems to mitigate drought impacts. The hedging rules were based on rationing supplies of demands when the water volume stored in the system at a given time was lower than the activation thresholds. Their methodology established a two-phase method combining PS with the simulation of the water system which consists of jointly operated reservoirs. The hedging rules were based on monthly storage levels that triggered restrictions on the demands. They successfully applied their model to four water resource systems in Spain.

6 Harmony Search (HS)

HS was proposed by Geem et al. (2001) as a stochastic random search optimization algorithm. It was conceptualized as the musical process searching for perfect state of harmony. Decison variable, values of decision variables, optimization solution vector, and solution iterations are analogous to each musician, collection of notes in memories, harmonies, and improvisations, respectively (Geem et al. 2001). HS searches optimal solution by considering multiple solution vectors as in the other evolutionary based methods. However, reproduction process in HS is different than those in others. For example; in GA, a new offspring is generated from two parents in the population. In DE, the off-spring is generated from a single parent. However, in HS, the offspring is generated from all of the existing vectors stored in the harmony memory (HM).

6.1 HS Algorithm Basics

The algorithm steps can be simple summarized as follows (Karahan et al. 2013):

Step 1. Random vectors ($x^1 \ldots \ldots x^{HMS}$) are generated as many as harmony memory size (HMS) and then their fitness values, f(x), are computed and stored in harmony memory (HM), which has the following form:

$$HM = \left[\begin{array}{ccccc|c} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 & f(x^1) \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 & f(x^2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} & f(x^{HMS-1}) \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} & f(x^{HMS}) \end{array}\right] \quad (8)$$

where f(x) is the objective function, x is the decision variable, and N is number of decision variables (Atrabi et al. 2015).

Step 2. New vector x' is generated. For each component x_1',

- pick the stored value from HM, $x_1' = x_1^{\mathrm{int}\left(rand(0,1)^* HMS\right)+1}$ with the probability of harmony memory considering rate (HMCR),
- pick a random value within the allowed range with probability (1-HMCR).

Step 3: If the value in Step 2 comes from HM, then

- change x_1' by a small amount: $x_1' = x_1' + bw^* rand(0,1)$ for continuous variable, with probability of pitch adjusting rate (PAR) or do nothing with probability (1 − PAR).

Step 4: Replace x^{Worst}with x' if x' is better than the worst vector x^{Worst} in HM.

Step 5: Repeat from Step 2 to Step 4 until termination criterion (e.g. number of maximum iterations) is satisfied.

Fig. 9 shows the flowchart for HS algorithm. The details of HS can be obtained from the literature (Geem et al. 2001; Karahan et al. 2013; Atrabi et al. 2015).

6.2 HS Algorithm Applications

We see first applications of HS in the area of water resources engineering by the work of Kim et al. (2001) who sucessfully estimated optimal values of the parameters of the nonlinear Muskingum model. Then, Geem (2006) used HS algorithm for the optimal design of water distribution network. He/she minimized the cost while

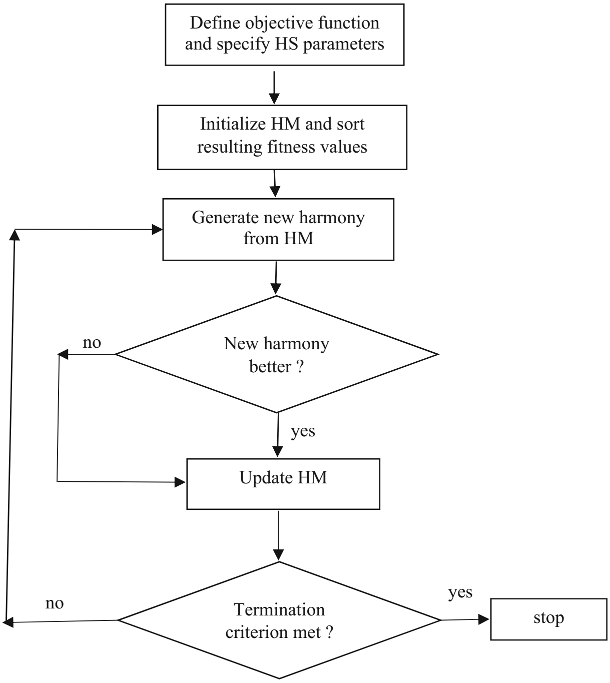

Define objective function and specify HS parameters
Initialize HM and sort resulting fitness values
Generate new harmony from HM
New harmony better ?
no
yes
Update HM
Termination criterion met ?
no
yes
stop

Fig. 9 Flowchart for HS algorithm

optimizing the design. Ayvaz (2007) combined HS optimization algorithm with numerical model solving groundwater flow governing equations to detrmine aquifer prameters and zone structures. HS algorithm well captured the true transmissivity field for different HS solution parameters. He/she concluded that the meta-heuristic HS algorithm is simpler and faster than GA. On the other hand, as the termination criterion becomes more sensitive, the increase in the computational burden can be observed.

Karahan et al. (2013) also employed HS in obtaining optimal values of the parameters of the nonlinear Muskingum model. In their methodology, they imposed an indirect penalty function on the model to prevent negativity of outflows and storages. They were able to find the global or near-global minimum regardless of the initial parameter values with fast convergence. Ayvaz (2009) combined MODFLOW, numerially solving three dimensional groundwater flow, with optimization algorithm of HS for the optimal management of groundwater resources. The optimal mangement included the maximization of total pumping from an aquifer, and minimization of the total pumping cost to satisfy the given demand. He/she concluded that solving problems with large number of decision variables by HS may not be feasible due to the requirement of large number of simulations to find the optimum solution. For such problems, he/she suggested the hybridization of the HS with a gradient-based optimization algorithm to reduce the number of simulations. According to the results of the sensitivity analysis in his/her work, although a change in HMCR and PAR parameters did have a minor effect on the final objective function value, the number of simulations required to obtain the optimum solution was significantly affected. In addition, the change in HMS affects the solution accuracy, i.e., the lower HMS may result in the premature convergence while the higher one cause an increase in the number of simulations.

7 Genetic Programming (GP)

Koza (1992) was the first to introduce GP which uses a tree structure including nodes and branches. GP is a random search heuristic method that mimics the process of natural evolution. It represents a complex phenomenon with mathematical and logical relations. GP was built on evolutionary methods, particularly on GA which is used to evolve the best values for a given set of model parameters. GP allows the basic structure of the model to evolve, together with the values of its parameters. In a simple GA, model parameter values are usually binary strings, whereas in GP algebraic expressions are used (Savic et al. 1999). GP uses parse trees instead of lines of code to represent hierarchical programs. The *root node* is the first element of the tree (e.g., x sign in Fig.10), the *interior nodes* are the functions (e.g., + sign in Fig.10) and the *leaf nodes* are the constants or the variables (e.g., a, c, 3.2 in Fig.10). Tree structures are capable of representing hierarchical programs of any complexity. Functions may include arithmetic operator (+, x, −), mathematical functions (sin, cos, log, exp), Boolean operators (and, or, not), logical expressions (if-then-else), iterative functions (do-until), or any other user-defined function (Savic et al. 1999). Each tree stand for a possible solution, i.e., it is analogous to chromosome in GA. The evaluation of tree in objective function results in its fitness.

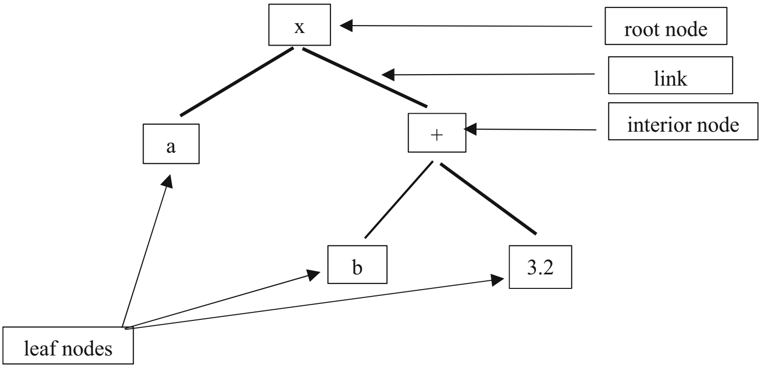


Fig. 10 Tree representation of algebraic expression '*a x (b + 3.2)*' in GP algorithm

7.1 GP Algorithm Basics

The main steps of GP algorithm include initialization, selection, and reproduction (Mehdipour et al. 2013). Initial population is formed by randomly generating individual trees. It is possible to generate in the order of hundreds of trees although this number partly depends also on the problem at hand. Fitness values of the trees are evaluated and the selection is performed by eliminating the weaker ones using the roulette wheel or ranking methods (Orouji et al. 2014). Crossover and mutation operations are employed for the reproduction of new individuals to better search the solution space. To do so, selected pair of parent trees are subjected to crossover by randomly swaping sub-trees of the parent ones. After the crossover operation, the mutation can be used by randomly exchanging a nodal or function variable with another one. Fig. 11 shows the main operations of GP. Fig. 2 can also, more less, represent the flow chart for GP algorithm.

7.2 GP Algorithm Applications

The first application of GP in water resources engineering was performed by Savic et al. (1999) who employed it for flow prediction in Kirkton chatchment in Scotland. Unlike ANN, Savic et al. (1999) pointed out, GP gives an insight into the relationship between input (rainfall and evaporation) and output (runoff) data and partially alleviates the problem of identifying the large number of parameters necessary for conceptual model calibration.

Babovic and Keijzer (2000), by GP, determined empirical relationships for the the resistance to flow, due to vegitation. Liong et al. (2001) calibrated rainfall runoff model of Storm Water Management Model (SWMM) by using GP algorithm. Liong et al. (2001) stated that, in contrast to physically based models requiring vast amount of catchment characteristics data, GP requires only rainfall and runoff data set. In addition, GP based models are simple and readily implementable. Its computational time is very short which is significant in flood warning and evacuation management.

Using GP algorithm; Giustolisi (2004) determined Chezy resistance coefficient in corrugated channels, Kizhisseri et al. (2005) explored a better correlation between the temporal pattern of flow field and sediment transport by utilizing numerical model results and field data, Sivapragasam et al. (2008) routed flood hydrographs, Izadifar and Elshorbagy (2010) modeled evapotranspiration, Azamathulla et al. (2010) predicted bridge-pier scour, Azamathulla et al.

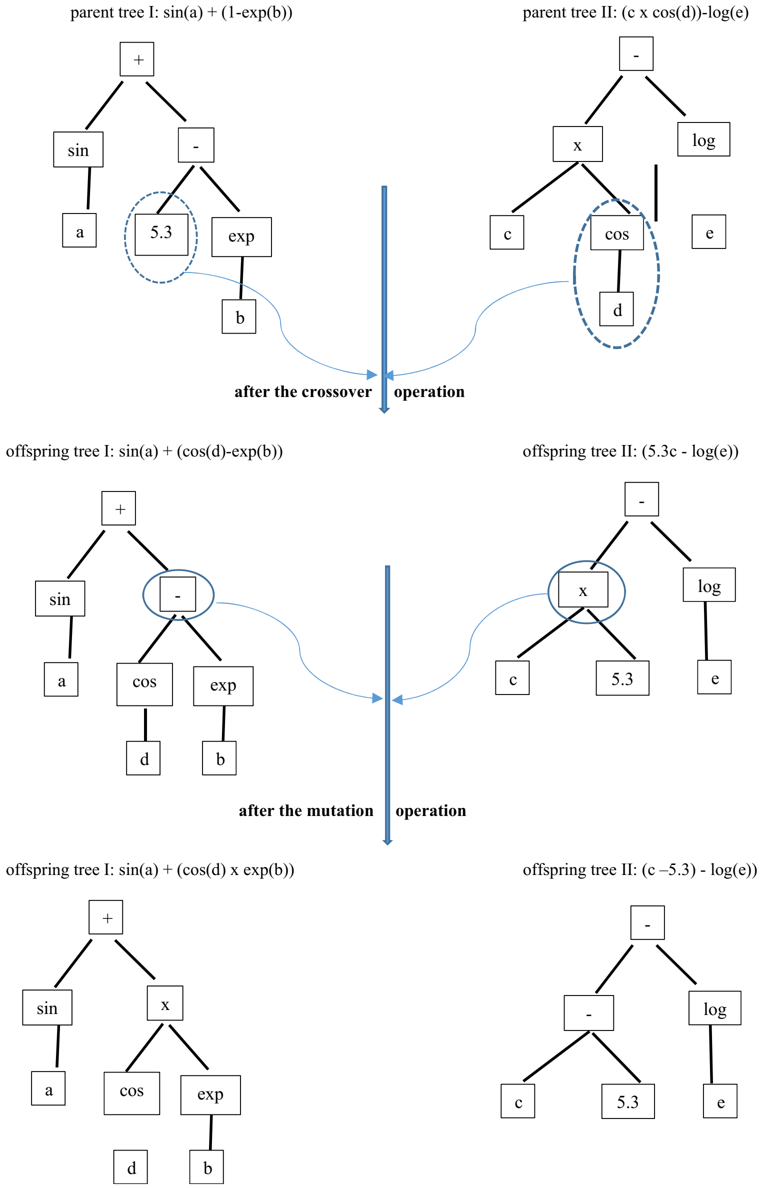

parent tree I: sin(a) + (1-exp(b))
+
sin
-
a
5.3
exp
b
parent tree II: (c x cos(d))-log(e)
-
x
log
c
cos
e
d
after the crossover
operation
offspring tree I: sin(a) + (cos(d)-exp(b))
+
sin
-
a
cos
exp
d
b
offspring tree II: (5.3c - log(e))
-
x
log
c
5.3
e
after the mutation
operation
offspring tree I: sin(a) + (cos(d) x exp(b))
+
sin
x
a
cos
exp
d
b
offspring tree II: (c –5.3) - log(e))
-
-
log
c
5.3
e

Fig. 11 Schematic representation of GP operators of crossover and mutation

(2011) produced stage-discharge relations for Pahang River in Malasia, and Azamathulla and Ghani (2011) predicted longitudinal dispersion coefficient in natural channels. The equations that they proposed as a result of GP applications however may not not make much sense. This is the typical problem in GP where the trees that involve many algebraic expressions may not much tell about the physics of the problem at hand. Mehdipour et al. (2013) applied GP to stage hydrograph routing as well. They compared their solutions against the numerical models that required substantial data on flow and channel characteristics. GP using less data was capable of routing the hydrographs. However, the same problem is encountered here too and that is the proposed equations by GP do not make much sense with regard to the physics of the flood routing process. The equations involve many algabraic expressions that are used by GP for the puspose of just optimal fitting.

Mehdipour et al. (2013) employed GP to extract operational decision rules in a reservoir system supplying downstream demand and generating hydropower energy. The extracted GP rules, which provided more accurate and appropriate decision variables in terms of computed objective function values, were superior to common linear and nonlinear operational ones. They concluded that GP is flexibile, efficient, and considerable accurate. Li et al. (2014) employed GP for identifying explicit operating rules for multipurpose reservoir system. They proposed the procedure in 4 steps. First, optimal operation trajectory of the multi-reservoir system was determined by the dynamic programming. Second, input variables of operating rules were selected by GP. Third, formulation of operating rules were identified by GP. Finally, key parameters of operating rules were refined by the parameterization-simulation-optimization method. The inflow and storage energy terms were selected as input variables for total output of the aggregated reservoir and for decomposition. They applied their model to multi-reservoir system in China that included the Three Gorges cascade hydropower reservoirs of Three Gorges and Gezhouba reservoirs and the Qing River cascade hydropower reservoirs of Shuibuya, Geheyan and Gaobazhou reservoirs. Their results revealed that power energy term could more effectively reflect the operating rules than water quantity for the hydropower systems. The derived operating rules were easier to implement for practical use and more efficient and reliable than the conventional operating rule curves. They also stated that the physical explanation of the identified GP formulation is not explicit, which needs further research.

Mehdipour et al. (2014) employed GP to extract governing groundwater flow equations in Ghaen and Karaj aquifers in Iran. Using a numerical model, they produced input-output data sets for the intelligence method to estimate water table variations. Several combinations of recharge and discharge were considered at each cell of the computational network and the corresponding water tables were calculated. Thus, the recharge and discharge data formed the input set while the hydraulic head was the output. Using these data sets, they extracted governing equations at each cell such that it was possible to simulate water table distribution in a portion of the aquifer, without modeling the entire aquifer network with less computational effort. However, they also stated that GP provides a non-physically-based analysis for natural phenomena. Hakimzadeh et al. (2014) collected dam breach data from literature, laboratory experiments and numerical model simulations to calibrate and validate GP to successfully simulate outflow hydrograph due to earthen dam breach. Kitsikoudis et al. (2015) used GP for the predciton of for the mean flow velocity prediction in sand-bed and gravel-bed rivers. They showed that GP was robust.

8 Gene Expression Programming (GEP)

Gene-Expression Programming (GEP), which is a variant of GA and GP, was developed by Ferreira (2001). While the genome is encoded in simple linear chromosomes of fixed length, the genes are expressed as a phenotype in the form of expression trees.

8.1 GEP Algorithm Basics

The process in GEP is similar to other evolutionary algorithms, in which, initial population is randomly created, their fitness values are computed, the strong ones are selected, and offsprings are generated. For creating new offsprings, the genetic opeartions involve mutation, inversion, transposition, and recombination (Azamathulla et al. 2011). Mutation is the most effeicient and the main opearator that sometimes it can be solely used (Ferreira 2006).

Mutation can be applied to any position on genome provided that structure of the chromosome is not altered., i.e., the length stays the same, any function can be replaced by a terminal function in the heads, and the terminals can only be changed into other terminals in the tail. Inversion operator selects and inverts a sequence within the head of gene (Azamathulla et al. 2011). Insertion Sequence (IS) and Root Insertion Squence (RIS) are two different transposition elements. IS elements are short genomic segments having a function or terminal at the first position. This operator randomly chooses a chromosome, a gene to be modified, the start and end of the IS element, which is then transposed to the start of the gene just after the root (Ferreira 2001). The only difference in RIS is that the starting point is always a function. In gene transposition, entire gene acts as a transposion and it transposes itself to the beginning of the chromosome. In contrast to the other forms of transposition, in gene transposition, the transposon (the gene) is deleted at the place of origin (Ferreira 2001). As single and double, there are two crossover/recombination methods. In single crossover, the parent chromosomes are paired at the same selected point. The portion of the gene downstream of the crossover point is subsequently exchanged between the two chromosomes. In double crossover, two parent chromosomes are paired and two points are randomly chosen as crossover points. The material between the crossover points is then exchanged between the parent chromosomes, forming two new daughter chromosomes (Azamathulla et al. 2011). In gene crossover, entire genes are exchanged between two parent chromosomes, forming two offspring chromosomes containing genes from both parents. The exchanged genes are randomly chosen and they occupy exactly the same positions as in the parent chromosomes. The details can be obtained from Ferreira (2001, 2006) and Azamathulla et al. (2011).

8.2 GEP Algorithm Applications

The first application of GEP in the area of water resources engineering was performed by Giustolisi (2004) who used it to determine Chezy resistance coefficient in corrugated channels. This study was followed by Eldrandaly and Negm (2008) who evaluated the performance of GEP for hydraulic data mining. Ghani and Azamathulla (2010) employed GEP for sediment transport in sewer systems, while Zakaria et al. (2010) used it for the estimation of total bed load.

Azamathulla et al. (2011) employed GEP for modeling the stage discharge relationship for the Pahang River. They compared the performance of GEP against the conventional methods of the stage rating curve (SRC) and regression techniques and concluded that GEP performs much better. They presented explicit formulations by GEP for the stage and discharge relations. The formulations involve many algabraic expressions such that the representation of the physical process by these are open to question. Fernando et al. (2012) employed GEP for multimodel combination of rainfall runoff models. They concluded that GEP combination modeling approach offers a promising way to seek flow forecasts through a transparent model combination method as opposed to the black-box approaches. Azamathulla and Jarrett (2013) used GEP to estimate roughness coefficient for high gradient streams, employing 75 discharge measurements on 21 high gradient streams in Colorado, USA. They observed that GEP automatically simplified genetic programs during an evolutionary processes (or evolves) to obtain the most robust computer program (e.g., simplify mathematical expressions, decision trees, polynomial constructs, and logical expressions). In addition, GEP clearly demonstrated the nonlinearity and complexity of flow in high gradient streams.

9 Concluding Remarks and Future Direction

Table 1 summarizes some distinctive characteristics of the evolutinary optimization algorithms, outlined above. In addition to modern optimization methods (MOMs) based on evolutionary search outlined in this paper, there have been, in recent years, also developments of new MOMs such as the cuckoo search, the honey-bees mating, the bee colony, the cat swarm, the weed, the artificial life, etc. Also, there are studies using the hybridization of one or more of these optimization algorithms. We recently see some applications of these optimization algorithms in the area of water resources engineering.

MOMs can handle functions that are nondifferentiable, nonlinear, and multimodal having complex search space with many local optimal solutions. Hence, they can be used in solving nonlinear, nonconvex, multimodal problems for which deterministic search techniques incur difficulty or fail completely, despite the fact that mathematical analysis of the evolutionary algorithms remains implicit. This is because, the interaction of various components in all such metaheuristic algorithms is highly nonlinear, complex, and stochastic. Careful selection of strategy and parameters is essential for a meaningful analysis of the problem. This requires extensive sensitivity analysis and assessment of the robustness of the methodologies.

MOMs make few or no assumptions about the problem being optimized and they can search very large spaces of candidate solutions. However, they do not guarantee an optimal solution. They mostly come in the vicinity of global optimum. To reach the exact global optimum point, the gradient decent method can further be employed. Proper selection of population size is essential to ensure the efficient optimization.

MOMs allow relatively simple coding, low computational cost, fast converges, and adaptiveness. They can overcome the problem of high computational efforts required when a conventional search techniques were used. On the other hand, they can require expensive computation when linked with numerical simulation models and this can limit their suitability for implicit and/or explicit stochastic optimization. Furthermore, when the system dimensions are increased, their flexibility, convergence ability, and coompuational efficiency can become a problem.

In contrast to physically based models requiring vast amount of data, MOMs require data mainly on the basic input-output variables. MOMs are, on the other hand, can not shed much light on the physics of the processes. Numerical models such as two-dimensional hydrodynamic models, CFD-3D models, and/or Integrated Catchment models need long computational (CPU) time, in the order of hours (even days), for a simulation of a real world case-study. One way of easing this problem could be the employment of the surrogate meta-models.

Table 1 Summary of some characteristics of the evalutionary optimization methods

Methods	Basic unit	Basic operators	Characteristics
GA	Chromosome	Fitness evaluation, selection, crossover, mutation	• Proper selection of population size is essential to ensure the efficient optimization. • It has extrapolation capability. • Computational time required for a GA_based linked simulation-optimization model is high.
AC	Individual (ant)	Cost evaluation, and positive reinforcement via the use of pheromone trails	• It requires expensive computations which can limit their suitability for implicit and/or explicit stochastic optimization. • Mathematical analysis of the algorithm remains implicit.
DE	Chromosome	Cost evaluation, selection, crossover, mutation	• Non-uniform crossover is used, by which off-spring vector parameters are taken more often from one parent. • Careful selection of strategy and parameters is essential for a meaningful analysis. • Its flexibility, convergance ability, and computational efficiency would be low when the system dimensions are increased. • It relies more on mutation.
PS	Particle (bird)	Position and velocity updating	• It has relatively simple coding, low computational cost, fast converges, and adaptiveness. • Evolution of generations occurs by cooperation and competition. • When its parameters are set right, it can achieve nearly global optimum solution in few iterations.
HS	Variable (musician)	Fitness evaluation and replacement.	• Off-spring is generated from a single parent among all of the existing vectors. • It can find the global or near-global minimum regardless of the initial parameter values with fast convergence. • It is simple but as the termination criterion becomes more sensitive, computational burden can increase. • Change in HMS can affect solution accuracy, i.e., the lower HMS may result in the premature convergence while the higher one cause an increase in the number of simulations.
GP	Tree (choromosome)	Fitness evaluation, selection, crossover, mutation	• It is simple, flexibile, effecient and readily implementable, and robust. • Its computational time is very short and it is accurate. • In GP where the trees that involve many algebraic expressions may not much tell about the physics of the problem at hand.
GEP	Choromosome	Fitness evaluation, selection, mutation, inversion, transposition, recombination (crossover)	• Mutation is the most effeicient and the main opearator that sometimes it can be solely used.

Although, various studies have attempted to carry out convergence analysis for some of MOMs, this point remains an active and challenging topic. Other active research directions can include the effective parallelization of algorithms and understanding and characterization of the behavior of algorithms and their convergence while solving a novel problems. The conjuctive use of the models can able to cope with the highly nonlinear system behaviour and local minimum problems. Their effective use could be accomplished when they are used in conjuction with physics based models. Future research can also be diverted to the area where the black box nature of MOMs could be disclosed to shed a light on the physics of the process being simulated.

References

Abbaspour KC, Schulin R, van Genuchten MT (2001) Estimating unsaturated soil hydraulic parameters using ant colony optimization. Adv Water Resour 24(8):827–841

Afshar A, Kazemi H, Saadatpour M (2011) Particle swarm optimization for automatic calibration of large scale water quality model (CE-QUAL-W2): Application to Karkheh Reservoir, Iran. Water Resour Manag 25: 2613–2632

Afshar A, Shojaei N, Sagharjooghifarahani M (2013) Multiobjective calibration of reservoir water quality modeling using multiobjective particle swarm optimization (MOPSO). Water Resour Manag 27:1931–1947

Afshar A, Massoumi F, Afshar A, Marino MA (2015) State of the art review of ant colony optimization applications in water resource management. Water Resour Manag 29(11):3891–3904

Afzal J, Noble DH, Weatherhead EK (1992) Optimization model for alternative use of different quality irrigation waters. J. Irrig Drain Eng 118(2):218–228

Atrabi HB, Qaderi K, Rheinheimer DE, Sharifi E (2015) Application of Harmony Search Algorithm to Reservoir Operation Optimization. Water Resour Manag 29:5729–5748

Ayvaz MT (2007) Simultaneous determination of aquifer parameters and zone structures with fuzzy c-means clustering and meta-heuristic harmony search algorithm. Adv Water Resour 30:2326–2338

Ayvaz MT (2009) Application of harmony search algorithm to the solution of groundwater management models. Adv Water Resour 32(6):916–924

Ayvaz MT (2016) A hybrid simulation-optimization approach for solving the areal groundwater pollution source identification problems. J. Hydrology 538:161–176

Ayvaz MT, Karahan H (2008) A simulation/optimization model for the identification of unknown groundwater well locations and pumping rates. J. Hydrol 357(1–2):76–92

Azamathulla HM, Ghani AA (2011) Genetic programming for predicting longitudinal dispersion coefficients in streams. Water Resour Manag 25:1537–1544

Azamathulla HM, Jarrett RD (2013) Use of gene-expression programming to estimate Manning's roughness coefficient for high gradient streams. Water Resour Manag 27:715–729

Azamathulla HM, Ghani AA, Zakaria NA, Aytac G (2010) Genetic programming to predict bridge pier scour. J. Hydraul Eng 136(3):165–169

Azamathulla HM, Ghani AA, Leow CS, Chang CK, Zakaria NA (2011) Gene-expression programming for the development of a stage-discharge curve of the Pahang River. Water Resour Manag 25(11):2901–2916

Babovic V, Keijzer M (2000) Genetic programming as a model induction engine. J. Hydroinformatics 2(1):35–60

Bellman R (1957) Dynamic programming. Princeton University Press, Princeton

Chandramouli V, Raman H (2001) Multi-reservoir modeling with dynamic programming and neural networks. J. Water Resour Plan Manag 127(2):89–98

Chang F-J, Chen L (1998) Real-coded genetic algorithm for rule-based flood control reservoir management. Water Resour Manag 12:185–198

Chau KW (2007) A split-step particle swarm optimization algorithm in river stage forecasting. J Hydrol 34:131–135

Chiang P-K, Willems P (2015) Combine evolutionary optimization with model predictive control in real-time flood control of a river System. Water Resources Management 29:2527–2542

Crawley PD, Dandy GC (1993) Optimal operation of multiple-reservoir system. J. Water Resour Plan Manag 119(1):1–17

Dantzig G (1953) Notes on linear programming. RAND Corporation. (https://en.wikipedia.org/wiki/George_Dantzig)

Demotier S, Carlier J, Daguinos T, Kora R (2001) Using linear programming methods for optimizing the real-time pump scheduling. Dritan Nace, Bridging the Gap: 1–8

Dorigo M (1992) Optimization, learning and natural algorithms. Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy. (in Italian)

Dorigo M, Maniezzo V, Colorni A (1996) Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst., Man, Cybern., Part B: Cybern. 26(1):29–41

Eldrandaly K, Negm AA (2008) Performance evaluation of gene expression programming for hydraulic data mining. Int Arab J Inf Technol 5(2):126–131

Fernando AK, Shamseldin AY, Abrahart RJ (2012) Use of gene expression programming for multimodel combination of rainfall-runoff models. J Hydrol Eng 17(9):975–985

Ferreira C (2001) Gene expression programming: A new adaptive algorithm for solving problems. Complex Systems 13(2):87–129

Ferreira C (2006) Gene expression programming: Mathematical modeling by an artificial intelligence. 2nd Edition, Springer-Verlag, Germany

Gaur S, Sudheer C, Graillot D, Chahar BR, Kumar DN (2013) Application of artificial neural networks and particle swarm optimization for the management of groundwater resources. Water Resour Manag 27:927–941

Geem ZW (2006) Optimal cost design of water distribution networks using harmony search. Eng Optim 38(3): 259–280

Geem ZW, Kim J-H, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68

Ghani AA, Azamathulla HM (2010) Gene-expression programming for sediment transport in sewer pipe systems. J Pipeline Syst Eng Pract, ASCE 2(3). doi :10.1061/(ASCE)PS.1949–1204.0000076

Giustolisi O (2004) Using genetic programming to determine Chezy resistance coefficient in corrugated channels. J Hydroinformatics 6(3):157–173

Goldberg DE (1983) Computer-aided gas pipeline operation using genetic algorithms and rule learning. PhD Thesis. University of Michigan, Ann Arbor, MI, USA

Goldberg DE (1999) Using time efficiently: Genetic-evolutionary algorithms and the continuation problem. In: Proceedings, Genetic and Evolutionary Computation Conference, pp: 212–219

Guitron A (1981) Hydro-electrical model for optimal operation of a single multipurpose reservoir. J. Hydrology 51(1–4):67–73

Gurarslan G, Karahan H (2011) A parameter estimation technique for the nonlinear muskingum flood routing model, 6th EWRA International Symposium-Water Engineering and Management in a Changing Environment, 2011, Catania, Italy

Gurarslan G, Karahan H (2015) Solving inverse problems of groundwater-pollution-source identification using a differential evolution algorithm. Hydrogeol J 23(6):1109–1119

Hakimzadeh H, Nourani N, Amini AB (2014) Genetic programming simulation of dam breach hydrograph and peak outflow discharge. J. Hydrol Eng 19(4):2014

Hall WA, Howell DT (1963) The optimization of single-purpose reservoir design with the application of dynamic programming to synthetic hydrology samples. J. Hydrology 1(4):355–363

Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Michigan

Ibanez KL, Prasad TD, Paechter B (2008) Ant colony optimization for optimal control of pump in water distribution networks. J. Water Resour Plan Manag 134(4):337–346

Imrie CE, Durucan S, Korre A (2000) River flow prediction using artificial neural networks: generalization beyond the calibration range. J Hydrol 233:138–153

Izadifar Z, Elshorbagy A (2010) Prediction of hourly actual evapotranspiration using neural network, genetic programming, and statistical models. Hydrol Process 24(23):3413–3425

Jalali MR, Afshar A, Marino MA (2005) Improved ant colony optimization algorithm for reservoir operation. Hydroinformatics Center, Civil Engineering Department, Iran University of Science and Technology, Tehran, Iran. (technical report)

Jha MK, Nanda G, Samuel MP (2004) Determining hydraulic characteristics of production wells using genetic algorithm. Water Resour Manag 18:353–377

Jia B, Simonovic SP, Zhong P, Yu Z (2016) A multi-objective best compromise decision model for real-time flood mitigation operations of multi-reservoir system. Water Resour Manag 30:3363–3387

Jothiprakash V, Arunkumar R (2013) Optimization of hydropower reservoir using evolutionary algorithms coupled with chaos. Water Resour Manag 27:1963–1979

Jothiprakash V, Shanthi G (2006) Single reservoir operating policies using genetic algorithm. Water Resour Manag 20:917–929

Jothiprakash V, Shanthi G, Arunkumar R (2011) Development of operational policy for a multi-reservoir system in India using genetic algorithm. Water Resour Manag 25:2405–2423

Jowitte PW, Germanopoulos G (1992) Optimal pump scheduling in water-supply networks. J. Water Resours Plan Manag 118(4):406–422

Kalita HM, Sarma AK, Bhattacharjya RK (2014) Evaluation of optimal river training work using GA based linked simulation-optimization approach. Water Resour Manag 28:2077–2092

Kantorovich LV (1939) Mathematical methods of organizing and planning production. Manag Sci, 6(4), 366–422. (July, 1960), pp. 366–422

Karaboga D, Okdem S (2004) A simple and global optimization algorithm for engineering problems: Differential evolution algorithm. Turkish J. Electr Eng 12(1):53–60

Karahan H (2011) Obtaining regional rainfall-intensity-duration-frequency relationship curves by using differential evolution algorithm. *Scientific Research Project of TUBITAK* (108Y299), Denizli, Turkey (In Turkish)

Karahan H (2012) Determining rainfall-intensity-duration-frequency relationship using particle swarm optimization. KSCE J Civ Eng 16(4):667–675

Karahan H, Ayvaz MT, Gurarslan G (2008) Determination of intensity-duration-frequency relationship by genetic algorithm: Case study of GAP. Teknik Dergi 19(2):4393–4407

Karahan H, Gurarslan G, Geem ZW (2013) Parameter estimation of the nonlinear Muskingum flood routing model using a hybrid harmony search algorithm. J. Hydrol Eng 18(3):352–360

Kareliotis SJ (1984) Optimization of a tree-like water-supply system. J. Hydrol 68(1–4):419–429

Kennedy J, Eberhart R (1995) Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks. pp. 1942–1948. doi:10.1109/ICNN.1995.488968

Kennedy J, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann. ISBN 1-55860-595-9

Kim JH, Geem ZW, Kim ES (2001) Parameter estimation of the nonlinear Muskingum model using harmony search. J. Am. Water Resources Assoc 37(5):1131–1138

Kitsikoudis V, Sidiropoulos E, Iliadis L, Hrissanthou V (2015) A machine learning approach for the mean flow velocity prediction in alluvial channels. Water Resour Manag 29:4379–4395

Kizhisseri AS, Simmonds D, Rafiq Y, Borthwick M (2005) An evolutionary computation approach to sediment transport modeling. In: Fifth international conference on coastal dynamics, Barcelona, Spain

Kolo DE, Haimes YY (1977) Capacity expansion and operational planning for regional water-resource systems. J. Hydrol 32(3–4):363–381

Koza JR (1992) Genetic programming on the programming of computers by means of natural selection. MIT Press, Cambridge

Kuczera G (1993) Network linear programming codes for water-supply headworks modeling. J. Water Resour Plan Manag 119(3):412–417

Kumar DN, Reddy MJ (2006) Ant colony optimization for multi-purpose reservoir operation. Water Resour Manag 20:879–898

Kumar DN, Reddy MJ (2007) Multipurpose reservoir operation using particle swarm optimization. J. Water Resour Plan Manag, ASCE 133(3):192–201

Lall U, Lin YC (1991) A groundwater management model for Salt Lake County, Utah with some water rights and water quality considerations. J. Hydrol 123(3–4):367–393

Li S, Liu Y, Yu H (2006) Parameter estimation approach in groundwater hydrology using hybrid ant colony system, Irwin (Eds.): ICIC 2 006, LNBI 4115, 182–191

Li X, Liu H, Yin M (2013) Differential evolution for prediction of longitudinal dispersion coefficients in natural streams. Water Resour Manag 27:5245–5260

Li L, Liu P, Rheinheimer DE, Deng C, Zhou Y (2014) Identifying explicit formulation of operating rules for multi-reservoir systems using genetic programming. Water Resour Manag 28:1545–1565

Liong S, Nguyen V, Gautam T, Wee L (2001) Alternative well calibrated rainfall-runoff model: genetic programming scheme. In: Brashear RW, Maksimovic C (eds) Urban drainage modeling, Proceedings of Symposium on Urban Drainage Modeling. 2001 World Water and Environmental Resourcess Congress, 20–24 May 2001, pp 777–787

Loucks DP, Stedinger JR, Haith DA (1981) Water resources systems planning and analysis. Prenctice Hall, Eaglewood Cliffs, New Jersey

Maier HR, Simpson AR, Zecchin AC, Foong WK, Phang KY, Seah HY, Tan CL (2003) Ant colony optimization for design of water distribution systems. J. Water Resour Plan Manag, ASCE 129:200–209

Marino MA, Mohammadi B (1983) Reservoir operation by linear and dynamic programming. J. Water Resour Plan Manag 109(4):303–319

Massoumi F, Afshar A, Afshar A, Marino MM (2015) State of the art review of ant colony optimization applications in water resource management. Water Resour Manag 29(11):3891–3904

McKerchar AI (1975) Optimal monthly operation of interconnected hydroelectric power storages. J. Hydrol 25(1–2):137–158

McKinney DC, Lin M-D (1994) Groundwater optimization using genetic algorithms. Water Resour. Res 30(6):1897

Mehdipour EF, Haddad OB, Orouji H, Marino MA (2013) Application of genetic programming in stage hydrograph routing of open channels. Water Resour Manag 27:3261–3272

Mehdipour EF, Haddad OB, Marino MA (2014) Genetic programming in groundwater modeling. J. Hydrol Eng, ASCE, ISSN 1084–0699/04014031(13)

Moghaddam A, Behmanesh J, Farsijani A (2016) Parameters estimation for the new four-parameter nonlinear Muskingum model using the particle swarm optimization. Water Resources Management 30:2143–2160

Needham JT, Watkins DW Lund JR, Nanda SK (2000) Linear programming for flood control in the Iowa and Des Moines rivers. J. Water Resour PlanManag, 126(3), 118–127

Nemhauser GL (1966) Introduction to dynamic programming. John Wiley & Sons Inc., New York 1966

Orouji H, Bozorg Haddad OB, Mehdipour EF, Mariño MA (2014) Flood routing in branched river by genetic programming. Proceedings of the Institution of Civil Engineers-Water Management, 167(2), 115–123

Ostadrahimi L, Marino MA, Afshar A (2012) Multi-reservoir operation rules: Multi-swarm PSO-based optimization approach. Water Resour Manag 26:407–427

Ostfeld A (2011) Ant colony optimization for water resources analysis- review and challenges. Chapter 11 in " Ant colony optimization- methods and applications", InTech. publishing, 342 pages

Pasha MFK, Lansey K (2009) Optimal pump scheduling by linear programming. Proceedings of the World Environmental and Water Resources Congress, Kansas City, 17–21 May 2009, pp 1–10

Perea RG, Poyato EC, Montesinos P, Diaz JAR (2016) Optimization of irrigation scheduling using soil water balance and genetic algorithms. Water Resour Manag 30:2815–2830

Price K, Storn R (2005) Home page of differential evolution, URL:http://www.icsi.Berkeley.edu/~storn/code.html

Price K, Storn RM, Lampinen JA (2005) Differential evolution: A practical approach to global optimization. Springer. ISBN 978-3-540-20950-8

Raju KS, Kumar DN (2004) Irrigation planning using genetic algorithms. Water Resour Manag 18:163–176

Reshma T, Reddy KV, Pratap D, Ahmedi M, Agilan V (2015) Optimization of calibration parameters for an event based watershed model using genetic algorithm. Water Resour Manag 29:4589–4606

Sahay RR (2012) Predicting transient storage model parameters of rivers by genetic algorithm. Water Resour Manag 26:3667–3685

Savic AD, Walters AG, Davidson JW (1999) A genetic programming approach to rainfall-runoff modeling. Water Resour Manag 13:219–231

See L, Openshaw S (2000) A hybrid multi-model approach to river level forecasting. Hydrol Sci J 45(4):523–536

Sen Z (2004) Genetic algorithm and optimization methods. Su Vakfı Yayınları, Istanbul, Turkey. (in Turkish)

Shi Y, Eberhart R (1998) A modified particle swarm optimizer. Proceedings of the IEEE international conference on evolutionary computation, Anchorage, Alaska, 69–73

Shourian M, Mousavi SJ, Tahershamsi A (2008) Basin-wide water resources planning by integrating PSO algorithm and MODSIM. Water Resour Manag 22:1347–1366

Singh A (2012) An overview of the optimization modelling applications. J. Hydrology 49(6–7):167–182

Singh VP, Woolhiser DA (2002) Mathematical modeling of watershed hydrology. J. Hydrol Eng 7(4):270–292

Sivapragasam C, Maheswaran R, Venkatesh V (2008) Genetic programming approach for flood routing in natural channels. Hydrol Process 22(5):623–628

Spiliotis M, Mediero L, Garrote L (2016) Optimization of hedging rules for reservoir operation during droughts based on particle swarm optimization. Water Resour Manag. doi:10.1007/s11269–016-1285-y in press

Storn R (1996) On the usage of differential evolution for function optimization. Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS). pp. 519–523

Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Glob Optim 11:341–359

Tao T, Lennox WC (1991) Reservoir operations by successive linear programming. J. Water Resour Plan Manag 117(2):274–280

Tayfur G (2009) GA-optimized model predicts dispersion coefficient in natural channels. Hydrol Res 40(1):65–78

Tayfur G (2012) Soft computing in water resources engineering: Artifical neural networks, fuzzy logic, and genetic algorithm. WIT Press, Southampton

Tayfur G, Karimi Y (2014) Use of principal component analysis in conjunction with soft computing methods for investigating total sediment load transferability from laboratory to field scale. Hydrol Res 45(4–5):540–550

Tayfur G, Moramarco T (2008) Predicting hourly-based flow discharge hydrographs from level data using genetic algorithms. J. Hydrology 352(1–2):77–93

Tayfur G, Singh VP (2011) Predicting mean and bankfull discharge from channel cross-sectional area by expert and regression methods. Water Resour Manag 25:1253–1267

Tayfur G, Barbetta S, Moramarco T (2009) Genetic algorithm-based discharge estimationat sites receiving lateral inflows. J. Hydrol Eng 14(5):463–474

Tayfur G, Karimi Y, Singh VP (2013) Principle component analysis in conjuction with data driven methods for sediment load prediction. Water Resour Manag 27:2541–2554

Tu Q, Li H, Wang X, Chen C (2011) Ant colony optimization for the design of small-scale irrigation systems. Water Resour Manag 25:1537–1544

Vasan A (2005) Studies on advanced modeling techniques for optimal reservoir operation and performance evaluation of an irrigation system. PhD thesis, Birla Institute of Technology and Science, Pilani, India

Vasan A, Raju KS (2004) Comparison of differential evolution and simulated annealing for reservoir system optimization: a case study in Rajasthan. National Symposium on Hydrology with Focal Theme on Water Quality, Roorkee, India, pp: 51–58

Vasan A, Raju KS (2007) Application of differential evolution for irrigation planning: An Indian case study. Water Resour Manag 21:1393–1407

Wang QJ (1991) The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. Water Resour. Res 27(9):2467

Zakaria NA, Azamathulla HM, Chang CK, Ghani A (2010) Gene-expression programming for total bed material load estimation—a case study. Sci Total Environ 408(21):5078–5085

Zhao T, Zhao J, Yang D (2014) Improved dynamic programming for hydropower reservoir operation. J. Water Resour Plan Manag 140(3):365–374

Zucco G, Tayfur G, Moramarco T (2015) Reverse flood routing in natural channels using genetic algorithm. Water Resour Manag 29:4241–4267