UC Santa Barbara
UC Santa Barbara Previously Published Works

Title

Application of the gravity search algorithm to multi-reservoir operation optimization

Permalink

https://escholarship.org/uc/item/84k1c1kv

Authors

Bozorg-Haddad, Omid
Janbaz, Mahdieh
Loáiciga, Hugo A

Publication Date

2016-12-01

DOI

10.1016/j.advwatres.2016.11.001

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Advances in Water Resources 98 (2016) 173–185

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

Application of the gravity search algorithm to multi-reservoir operation optimization

Omid Bozorg-Haddad[a,*], Mahdieh Janbaz[b], Hugo A. Loáiciga[c]

[a] Department of Irrigation & Reclamation Engineering, Faculty of Agricultural Engineering & Technology, College of Agriculture & Natural Resources, University of Tehran, Karaj, Tehran, Iran
[b] Department of Irrigation & Reclamation Engineering, Faculty of Agricultural Engineering & Technology, College of Agriculture & Natural Resources, University of Tehran, Karaj, Tehran, Iran
[c] Department of Geography, University of California, Santa Barbara, CA, United States

ARTICLE INFO

Article history:
Received 30 April 2016
Revised 27 October 2016
Accepted 2 November 2016
Available online 3 November 2016

Keywords:
Gravity search algorithm
Optimization
Multi-reservoir operation

ABSTRACT

Complexities in river discharge, variable rainfall regime, and drought severity merit the use of advanced optimization tools in multi-reservoir operation. The gravity search algorithm (GSA) is an evolutionary optimization algorithm based on the law of gravity and mass interactions. This paper explores the GSA's efficacy for solving benchmark functions, single reservoir, and four-reservoir operation optimization problems. The GSA's solutions are compared with those of the well-known genetic algorithm (GA) in three optimization problems. The results show that the GSA's results are closer to the optimal solutions than the GA's results in minimizing the benchmark functions. The average values of the objective function equal 1.218 and 1.746 with the GSA and GA, respectively, in solving the single-reservoir hydropower operation problem. The global solution equals 1.213 for this same problem. The GSA converged to 99.97% of the global solution in its average-performing history, while the GA converged to 97% of the global solution of the four-reservoir problem. Requiring fewer parameters for algorithmic implementation and reaching the optimal solution in fewer number of functional evaluations are additional advantages of the GSA over the GA. The results of the three optimization problems demonstrate a superior performance of the GSA for optimizing general mathematical problems and the operation of reservoir systems.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

The scarcity of water and recurrent drought in many parts of the world demand the careful management of water resources. Optimization methods are well suited for improving water resources planning and management. Papers dealing with optimization methods in reservoir operation can be classified as classic or evolutionary (Ahmadi et al., 2014; Ashofteh et al., 2013; 2015). The classical optimization methods are useful in finding the optimal solution of unconstrained maxima or minima of continuous and differentiable functions (Wehrens et al., 2006). Random search technique, linear programming, and dynamic programming fall in this category of classic optimization methods. Among the classical optimization methods Tilmant and Kelman (2007) reported a methodology for analyzing trade-offs and risks associated with large-scale water resources projects under hydrologic uncertainty. Their proposed methodology relies on the stochastic dual dynamic programming (SDDP) model to derive monthly or weekly operating rules for multipurpose multi-reservoir systems taking into account the stochasticity of the inflows, irrigation water withdrawals, minimum/maximum flow requirements for navigation, fishing and/or for ecological purposes. Ficchi et al. (2015) reported the improvement of the operation of a four-reservoir system in the Seine River basin, France, employing deterministic and ensemble weather forecasts and real-time control. Their simulation results demonstrated that the proposed real-time control system largely outperforms the no-forecasts management strategy, and that explicitly considering forecast uncertainty through ensembles can compensate for the loss in performance due to forecast inaccuracy.

The classic optimization methods, however, do not perform well in many complex problems with multimodal functions. Therefore, evolutionary methods (EM) were developed and are commonly used to tackle such problems. Evolutionary methods can handle any type of objective function, including multi-objective variants, and avoid entrapment in suboptimal solutions much more effectively than classical methods.

Several mathematical benchmark functions have been implemented for testing classic and evolutionary methods. Yao et al. (1999) proposed the improved fast evolutionary programming (IFEP) employing mixing of different search operations (mutations)

* Corresponding author.
E-mail addresses: OBHaddad@ut.ac.ir (O. Bozorg-Haddad), Ma_Janbaz@ut.ac.ir (M. Janbaz), Hugo.Loaiciga@ucsb.edu (H.A. Loáiciga).

http://dx.doi.org/10.1016/j.advwatres.2016.11.001
0309-1708/© 2016 Elsevier Ltd. All rights reserved.

and fast evolutionary programming (FEP). They compared the results with classic evolutionary programming (CEP) by testing 19 benchmark functions. Their results obtained with the benchmark functions showed that the IFEP performed better than or as well as the FEP and CEP for most tested benchmark functions. Vesterstrom and Thomsen (2004) evaluated the performance of differential evolutionary (DE), particle swarm optimization (PSO), and evolutionary algorithms (EAs) on a suite of 34 widely used benchmark functions. The results from their study indicated that DE generally outperformed the other algorithms, although two noisy functions, DE and PSO were outperformed by the EA. Wang et al. (2011) compared a multi-tier interactive genetic algorithm (MIGA) and the GA to find the optimal solution to long-term reservoir operation.

The optimization of multi-reservoir operations has been tackled with evolutionary algorithms. There are different rules for reservoir operations systems (Oliveira and Loucks, 1997). Bolouri-Yazdeli et al. (2014) evaluated real-time operation rules in reservoir systems operation. Long-term reservoir operation in this study relies on a specific inflow time series, and it is assumed that similar time series will be observed in the future under steady climatic conditions. In contrast, real-time operation a reservoir relies on current system parameters that account for reservoir inflow, downstream water demand, and the storage at the beginning of each operational period.

Mousavi et al. (2004) solved reservoir optimization with fuzzy and non-fuzzy stochastic dynamic programming (SDP). Celeste and Billib (2009) assessed the performance of seven stochastic models to determine optimal reservoir operating policies. Their models were based on implicit stochastic optimization (ISO) and explicit stochastic optimization (ESO), and on the parameterization–simulation–optimization (PSIO) approach. The ISO models include multiple regressions, two-dimensional surface modeling and a neuro-fuzzy strategy. Wardlaw and Sharif (1999) evaluated several alternative formulations of a genetic algorithm for reservoir systems by using a four-reservoir, deterministic, finite-horizon problem and a ten-reservoir problem. They found that the genetic algorithm approach is easily applied to complex systems and it was a viable alternative to stochastic dynamic programming approaches. Côté and Leconte (2015) compared stochastic optimization algorithms for hydropower reservoir operation. Three explicit stochastic optimization approaches, i.e., stochastic dynamic programming, sampling stochastic dynamic programming, and a scenario tree approach were compared with the operation of the Rio Tinto Alcan (RTA) hydropower system in Québec, Canada. Ahmadianfar et al. (2015) introduced the improved bat algorithm (IBA) with a hybrid mutation strategy to improve its global searching capacity.

Bozorg-Haddad et al. (2010) evaluated the performance of EAs in reservoir operation by testing the honey-bee mating optimization (HBMO) algorithm with three benchmark multi-reservoir (four and ten-reservoir) operation problems in the discrete and continuous domains. In addition, they compared the HBMO's results with those of the genetic algorithm (GA) and linear programming (LP). The comparison of the results of HBMO, GA, and LP demonstrated a superior performance of the HBMO. Bozorg-Haddad et al. (2015) applied the biogeography-based optimization (BBO) algorithm to solve reservoir operation problems. The BBO algorithm was verified with the optimization of three mathematical benchmark problems and then was applied to the operation optimization of a single reservoir system and a four-reservoir system. The performance of the BBO algorithm was compared with that of the genetic algorithm (GA) in solving the three optimization problems. Their results from five test problems demonstrated the superior capacity of the BBO to optimize general mathematical problems and the operation of reservoir systems. The genetic algorithm (GA) is one of the most widely used evolutionary algorithms this being the reason for using it as a comparison optimization method.

The GSA (gravity search algorithm) is an evolutionary method which is based on Newton's law of gravity (Rashedi et al., 2009). Newton's law states that particles attract each other with a force which is directly proportional to the product of their masses and inversely proportional to the square of the distance between the particles. Rashedi et al. (2009) compared the GAS with particle swarm optimization (PSO) and central force optimization (CFO) using well-known benchmark functions. Their results established the excellent performance of the GSA in solving various nonlinear functions. Ghalambaz et al. (2011) presented a hybrid neural network and gravitational search algorithm (HNGSA) method to solve the well-known Wessinger's equation which is a fully implicit first order nonlinear differential equation. Their results showed that the HNGSA produced a closer approximation to the analytic solution than other numerical methods and that it could easily be extended to solve a wide range of problems. Jadidi et al. (2013) proposed a flow-based anomaly detection system and applied a multi-layer perceptron (MLP) neural network with one hidden layer for solving it. The latter authors optimized interconnection weights of a MLP network with a gravitational search algorithm (GSA) and the proposed GSA-based flow anomaly detection system (GFADS) was trained with a flow-based data set. Chen et al. (2014) proposed an improved gravitational search algorithm (IGSA) and solved the identification problem for a water turbine regulation system (WTRS) system under load and no-load running conditions.

The GSA algorithm has not been previously applied to multi-reservoir operation systems. This paper tests the GSA algorithm with several well-known benchmark functions presented in the Appendix. The GSA algorithm is herein also applied to optimally operate: (a) a hydropower production problem, and (b) a well-known four-reservoir operation problem. The GSA is herein compared with the GA employing LP and NLP globally optimal solutions to test its capacity for solving long-term reservoir operation with single- and multi-reservoir systems. The GA was the first EA and its good performance is well established. The GA is available in the Matlab software thus making a relatively simple benchmark to implement in evaluating the performance of the other algorithms.

2. The gravitational search algorithm (GSA)

The GSA is a novel evolutionary algorithm. According to the GSA every particle or mass in a system determines the position and state of other particles according to the law of gravity (Rashedi et al. 2009). Each particle exhibits simple behavior In the GSA, and all of them follow intelligent pathways towards the near-optimal solution. For better understanding of the GSA one must resort to the law of gravity. Nature encompasses three type of masses: (1) active gravity mass, in which the gravity force increases with increasing mass; (2) passive gravity mass, in which the gravity force does not increase with increasing mass, and (3) inertial mass, that expresses the resistance of mass to change its position and movement. Particles in a system attract each other with a specific gravity force that is directly related to the product of their masses of the particles and inversely related to the square distance between any two particles (see Fig. 1):

$$F = G\frac{M_1 M_2}{R^2} \tag{1}$$

where F = gravity force (N); M_1 = active gravity mass (kg) of first particle; M_2 = passive mass (kg) gravity of second particle; G = Newton's gravitational constant [$(Nm^2)/kg^2$], and R distance separating the two particles (m). G is a parameter in the GSA that controls the searching capacity and the algorithm's efficiency. On the one hand, the searching capacity of the optimization algorithm increases whenever G increases. On the other hand, the

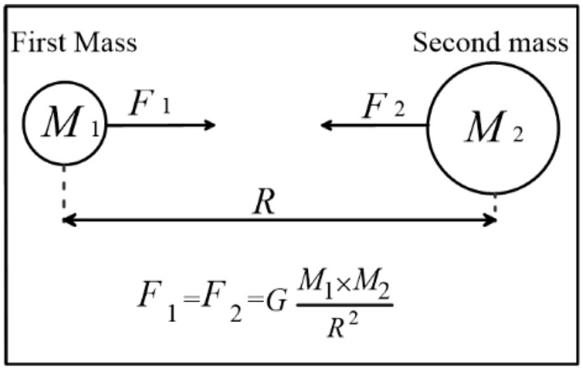


Fig. 1. Relations between particle mass and gravity force.

convergence efficiency of the search algorithm increases when G decreases. For these reasons, it is recommended to use a value of G that is set initially high and decreases with increasing time (Rashedi et al. 2009). A suitable formula for G is:

$$G(t) = G_0\, e^{-\frac{\alpha\, t}{T}} \qquad t = 1, 2, \ldots, T \tag{2}$$

where $G(t)$= Newton's gravitational constant as a function of the iteration number; $G_0$and α = controlling coefficients of the GSA; T = lifetime of the system (total number of iterations); and t = number of iterations. $G(t)$ is initialized in Eq. (2) at the beginning of the optimization and is reduced over time to control the search accuracy.

Newton's second law states that when a force, F, is applied to a particle, its acceleration a, (m/s^2), depends only on the force and its mass, M:

$$a = \frac{F}{M} \tag{3}$$

where a = particle acceleration, and M = inertial mass. The variation of the velocity or acceleration of any mass is equal to the force acted on the system divided by the mass of inertia.

The vector of positions X_i (1) in a D-dimensional system at time $t = 1$ is defined as follows:

$$X_i(1) = (x_i^1(1), \ldots, x_i^d(1), \ldots, x_i^D(1)) \qquad i = 1, 2, \ldots, S; t = 1 \tag{4}$$

in which x_i^d=represents the position of the i-th mass in the d-th dimension at time (iteration) $t = 1$ (represents a particle), D = number of decision variables; d = counter of decision variables, $d = 1$, 2, …, D; i = counter of populations in each iteration (represents a mass), and S = number of populations.

The steps of the GSA, as described in Rashedi et al. (2009), are synthetized below and depicted in Fig. 2:

(1) Initialization: In the first iteration (or step) the values of x_i^d (as decision variable) are chosen randomly and extend over the entire solution space.
(2) Evaluation of the objective function (fit_i). The value of the objective function is calculated for each mass and is denoted by fit_i. The value of fit_i is calculated by inserting X_i in the objective function. The best and the worst values of the objective function are called *best* (t) and *worst* (t), respectively, and are then determined as follows (under minimization):

$$worst(t) = Maximum\, fit_i(t) \qquad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T \tag{5}$$

$$best(t) = Minimum\, fit_i(t) \qquad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T \tag{6}$$

(3) Calculation of the relative normalized objective function $M_i(t)$. The value of $M_i(t)$ increases with increasing value of the objective function (fit_i) as follows:

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \qquad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T \tag{7}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{S} m_j(t)} \qquad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T \tag{8}$$

where $m_i(t)$ = the normalized objective function of the mass i at time (iteration) t, and $M_i(t)$= normalized objective function of mass i at time (iteration) t.
(4) Update the gravitational constant and calculate inserted force on each particle. The gravitational constant is updated base on Eq. (2) and the force acting on mass i from mass j [this is an extension of Eq. (1)] as follows:

$$F_{ij}^d(t) = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon}\left[x_j^d(t) - x_i^d(t)\right]$$
$$j = 1, 2, \ldots, S;\ i = 1, 2, \ldots, S;\ i \neq j;$$
$$t = 1, 2, \ldots, T;\ d = 1, 2, \ldots, D \tag{9}$$

where $F_{ij}{}^d(t)$ = the force action on first mass i from second mass j in time (iteration) t and dimension d; $M_i(t)$ = normalized first mass i (passive gravitational mass); $M_i(t)$ = normalized second mass j (active gravity mass); $x_j^d(t) - x_i^d(t)$ is the distance between two particles of mass i and mass j in dimension d; ε = small positive constant, and $R_{ij}(t)$ = the Euclidian distance between two masses i and j:

$$R_{ij}(t) = ||X_i(t), X_j(t)||_2$$
$$= \sqrt{[x_i^1(t) - x_j^1(t)]^2 + [x_i^2(t) - x_j^2(t)]^2 + \ldots + [x_i^D(t) - x_j^D(t)]^2}$$
$$j = 1, 2, \ldots, S;\ i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T;\ i \neq j \tag{10}$$

The GSA algorithm is randomized by assuming that the total force acting on mass i in a dimension d is a randomly weighted sum of the D-th components of the forces exerted on other masses:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{S} rand_j F_{ij}^d(t) \qquad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T;$$
$$d = 1, 2, \ldots, D \tag{11}$$

where $F_i{}^d(t)$ = summation force on mass i *in* dimension d and time (iteration) t, and $rand_j$ = a random number with uniform distribution in the interval [0, 1] that introduces random properties to the GSA.
(5) Calculate the acceleration and velocity of each particle. The acceleration of each mass in dimension d is calculated based on the second law of motion as follows:

$$a_i^d(t) = \frac{F_i^d}{M_{ii}} \qquad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T;$$
$$d = 1, 2, \ldots, D \tag{12}$$

where M_{ii} = inertial mass i that equals M_i. Substituting the force from Eqs. (9) and (11) into Eq. (12) yields:

$$a_i^d(t) = G(t)\sum_{\substack{j=1\\ j\neq i}}^{S}\left\{rand_j \frac{M_j(t)}{R_{ij}(t) + \varepsilon}[x_j^d(t) - x_i^d(t)]\right\}$$
$$i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T;$$

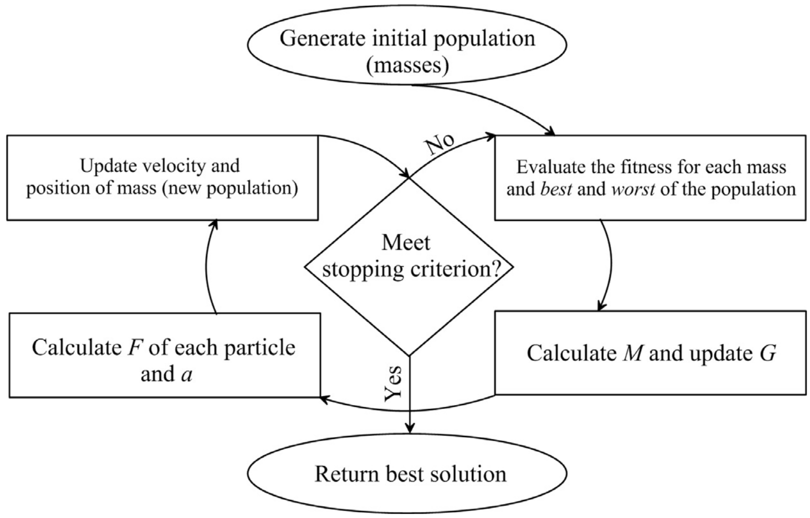


Fig. 2. The steps of the GSA.

$$d = 1, 2, \ldots, D \tag{13}$$

The velocity is calculated as follows:

$$v_i^d(t+1) = rand_i . v_i^d(t) + a_i^d(t) \quad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T;\ d = 1, 2, \ldots, D \tag{14}$$

where $v_i{}^d(t)$=velocity of mass i in dimension d and time (iteration) t, and $rand_i$=a uniform random variable in the interval [0, 1].

(6) Update position of masses. The new position of mass i in dimension d is given by:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T;\ d = 1, 2, \ldots, D \tag{15}$$

where $x_i{}^d(t)$=position of mass i in dimension d and time (iteration) t. The GSA algorithm generates the new X_i with the newly calculated positions of the masses.

(7) Apply the stopping criterion: Repeat Steps 2 to 6 until the stopping criterion is reached.
One way to improve the performance of GSA is to reduce the number of masses (which make up the population of solutions) with lapse of time in Eq. (11). Hence, a set of masses (defined as *Kbest*) with high merit are chosen as elite masses and apply their force to the other masses. *Kbest* is a function of time, in which the initial value decreases with time. In this manner all the masses apply a force initially, and *Kbest* is decreased linearly with the passage of time, until there is only one mass applying force to the other masses. Accordingly, Eqs. (11) and (13) are changed as follows:

$$F_i^d(t) = \sum_{j \in kbest, j \neq i} rand_j F_{ij}^d(t)\ i = 1, 2, \ldots, S;\quad t = 1, 2, \ldots, T;\ d = 1, 2, \ldots, D \tag{16}$$

$$a_i^d(t) = G(T) \sum_{\substack{j \in kbest \\ j \neq i}} \left\{ rand_j \frac{M_j(t)}{R_{ij}(t) + \varepsilon} [x_j^d(t) - x_i^d(t)] \right\} \quad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T;\quad d = 1, 2, \ldots, D \tag{17}$$

where *Kbest* is a set of elite masses with the best fitness values.

These previous 7 steps are implemented in a minimization problem. For a maximization problem Eqs. (8) and (9) must change to:

$$worst(t) = Minimum\,fit_i(t) \quad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T \tag{18}$$

$$best(t) = Maximum\,fit_i(t) \quad i = 1, 2, \ldots, S;\ t = 1, 2, \ldots, T \tag{19}$$

At the end of these steps the optimal values of the decision variable (x_i^d) and the objective function (fit_i) are obtained. Particles, masses, and the population in the GSA algorithm are shown in Fig. 3.

3. Multi-reservoir operation optimization rules

Three standard benchmarks functions were evaluated to test the GSA. Details about the test of the benchmark functions are provided in the Appendix. The capability of the GSA was also tested with the optimization of reservoir operation problems. Two reservoir operation problems were considered: one is a single reservoir that produces hydropower and minimizes the power deficit. The other is a four-reservoir problem that maximizes a benefit function. The schematic of the single reservoir is shown in Fig. 4. The quality of GSA results in terms of the process of achieving optimal solution in reservoir-operation have been obtained by comparing with the results of GA algorithm, a well-proven and reliable EA.

The general objective function includes minimization of the power-production deficit in the hydropower reservoir and the maximization of benefits (= minimization of negative benefits) in the four-reservoir system:

$$Minimize\, F = k \times \sum_{i=1}^{n} \sum_{t=1}^{T} \left(1 - \frac{P_i(t)}{PPC}\right)^2 - (1-k) \times \sum_{i=1}^{n} \sum_{t=1}^{T} a_i(t) \times R_i(t) \tag{20}$$

where $P_i(t)$=power generation (W) in period t=1, 2,…, T, and reservoir i=1, 2…, n; PPC = maximum power generation (W); R_i = reservoir release (m^3) in period t and reservoir i; $a_i(t)$=benefit function period t and Reservoir i; T=total number of periods; n=total number of reservoirs, and k=coefficients can be either 1 (for hydropower reservoir) or 0 (for four-reservoir). The power $P_i(t)$ (W) is calculated as follows:

$$P_i(t) = 9.81 \times 0.88 \times \frac{QR_i(t)}{0.2} \times \frac{(H_i(t) - 845)}{1000} \quad i = 1, 2, \ldots, n;$$

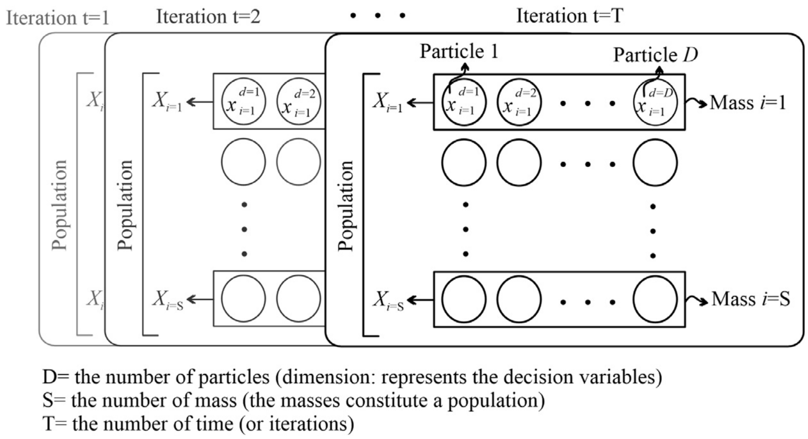


Fig. 3. Schematic of the "particles", "mass" and "time" in GSA algorithm.

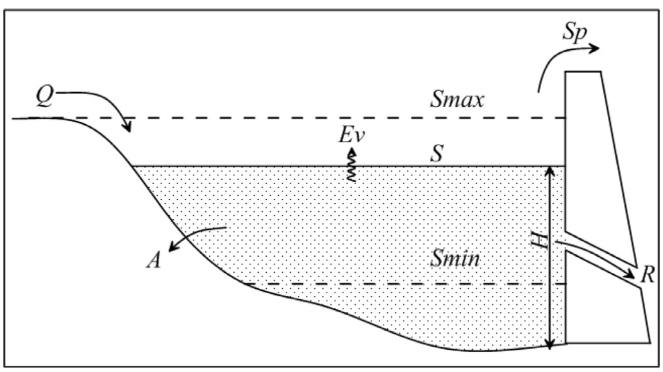


Fig. 4. Schematic of reservoir with inputs and outputs.

$$t = 1, 2, \ldots, T \quad (21)$$

where $QR_i\,(t)$ = turbine inflow (m^3/s) in period t and reservoir i; $H_i\,(t)$ = water level (m) in reservoir i and period t; 9.81 = acceleration of gravity (m/s^2); 0.88 = efficiency of power plant; 0.2 = plant functional coefficient and, 845 = reservoir tail water level (m), and 1000 is used for unit conversion. The decision variables in the objective function (20) are the reservoir releases $R_i(t)$.

$QR_i\,(t)$ is calculated as follows:

$$QR_i(t) = \frac{R_i(t)}{2.592} \quad i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \quad (22)$$

Reservoir area A (m^2) and reservoir water level H (m) formulas in terms of reservoir storage S (m^3) are as follows:

$$A_i(t) = \sum_{j=0}^{4} b_{ij} \times S_i(t)^j \quad i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \quad (23)$$

$$H_i(t) = \sum_{j=0}^{4} d_{ij} \times S_i(t)^j \quad i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \quad (24)$$

where b_{ij} and d_{ij} = area-volume coefficient and height-volume coefficient in Reservoir i, respectively.

The continuity constraint on reservoir storage is:

$$S_i(t+1) = S_i(t) + Q_i(t) - \left[\frac{(A_i(t) \times Ev_i(t))}{1000}\right] - M_{n\times n}R_i(t)$$
$$i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \quad (25)$$

where $S_i(t)$ = reservoir storage in period t and reservoir i; $Q_i(t)$ = reservoir inflow in period t and reservoir i (m^3); $A_i(t)$ = reservoir area in period t and reservoir i; $Ev_i(t)$ = reservoir evaporation

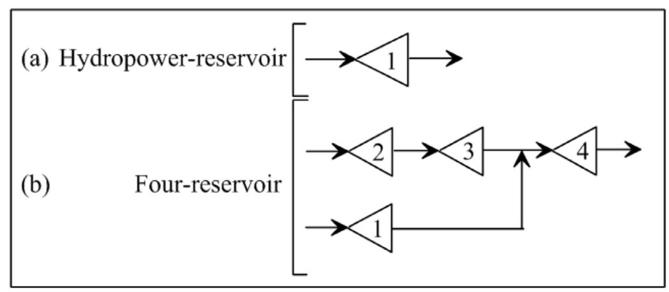


Fig. 5. Schematic representations of (a) hydropower-reservoir and (b) four-reservoir system.

in period t and Reservoir i (mm), (input and output of reservoir, see Fig. 4), and M = a n × n matrix of indices of reservoir connectivity of water releases and inflow (the relations between reservoirs based on inflow and outflow, see Fig. 5).

Releases from the reservoirs are constrained:

$$Rmin_i \le R_i(t) \le Rmax_i \qquad i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \quad (26)$$

where $Rmin_i$ = minimum release in Reservoir i (m^3), and $Rmax_i$ = maximum release in Reservoir i (m^3), $R_i(t)$ is modified as follows when $P_i(t)$ exceeds the PPC in the hydropower reservoir:

$$R_i(t) = \frac{PPC \times 0.2 \times 30 \times 24 \times 3600}{9810 \times 0.88 \times (H_i(t) - 845)}$$
$$i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \quad (27)$$

$R_i(t)$ in Eq. (27) was obtained by combining Eqs. (21) and (22). The constraint on reservoir storage is:

$$Smin_i(t) \le S_i(t) \le Smax_i(t) \quad i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \quad (28)$$

in which $Smin_i(t)$ = minimum storage in period t and Reservoir i (m^3), and $Smax_i(t)$ = maximum storage in period t and Reservoir i (m^3). Spillage is calculated when reservoir storage [$S_i(t)$] exceeds its maximum storage [$Smax_i(t)$] as follows:

$$Sp_i(t) = S_i(t) - \underset{i}{Smax} \Rightarrow S_i(t) = \underset{i}{Smax} \quad i = 1, 2, \ldots, n;$$
$$t = 1, 2, \ldots, T \quad (29)$$

where $Sp_i(t)$ = spillage in period t and Reservoir i (m^3).

Moreover, the initial storage of Reservoir i ($Sinitial_i$, m^3) is made equal to the ending storage of Reservoir i ($Starget_i$, m^3) (the periodicity constraint):

$$Sinitial_i = St\arg et_i \quad i = 1, 2, \ldots, n \quad (30)$$

If the reservoir storage does not meet the constraints (28) and (30), the result would be infeasible. Therefore, penalty functions must be added. The penalty functions, $Pe_i(t)$, are expressed as follows:

$$Pemax_i(t) = Pemax_i(t) + C_{max} \times [Smax_i(t) - S_i(t)]^2 \quad i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \tag{31}$$

$$Pemin_i(t) = Pemin_i(t) + C_{min} \times [Smin_i(t) - S_i(t)]^2 \quad i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \tag{32}$$

$$Petarget_i(t) = Petarget_i(t) + C_{target} \times [Starget_i - Sinitial_i]^2 \quad i = 1, 2, \ldots, n;\ t = 1, 2, \ldots, T \tag{33}$$

where C_{max}, C_{min}, and C_{target} = penalty coefficients for the violation of penalties on maximum storage ($Pemax$), minimum storage ($Pemin$) and inequality penalties on initial and target storage ($Petarget$), respectively. The penalized objective function becomes:

$$Minimize\, F = k \times \sum_{i=1}^{n}\sum_{t=1}^{T}(1 - \frac{P_i(t)}{PPC})^2 - (1-k) \times \left[\sum_{i=1}^{n}\sum_{t=1}^{T} a_i(t) \times R_i(t) - \sum_{i=1}^{n}\sum_{t=1}^{T} Pemax_i(t) - \sum_{i=1}^{n}\sum_{t=1}^{T} Pemin_i(t) - \sum_{i=1}^{n}\sum_{t=1}^{T} Petarget_i(t)\right] \tag{34}$$

where $k=1$ is for single reservoir for hydropower operation objective function and $k=0$ is for the four-reservoir operation objective function

4. Case studies

4.1. Single reservoir operation for power production

The system consists of one reservoir for power production as shown in Fig. 5(a), where reservoir releases are used for hydropower generation. The data used in this problem are based on data of the Karon4 hydropower dam in Iran. The objective function is to maximize power generation over a 5-year period or 60 operating periods (months) (or minimize power deficit) as written in Eq. (34) with $k=1$. The final storage in Reservoir 1 equals the beginning storage, and the PPC is expressed in 1000 watt$\times 10^3$ (MW).

The matrix of indices of reservoir hydraulic connectivity for the hydropower-reservoir problem is:

$$M_{1\times 1} = [-1] \tag{35}$$

Eq. (35) indicates an output from the hydropower reservoir when there are not inputs from other reservoirs. Reservoir inflow, Q, and reservoir evaporation, Ev, over the 60 operating periods are listed in Table 1.

Constraints posed on reservoir releases, R, and reservoir storages, S, (in 10^6m3) are:

$$0 \le R_1(t) \le 450 \qquad t = 1, 2, \ldots, 60 \tag{36}$$

$$1441.29 \le S_1(t) \le 2190 \qquad t = 1, 2, \ldots, 60 \tag{37}$$

The penalties C_{max}, C_{min}, and C_{target} are equal to 0, 50, and 50, respectively. The b_{1j} (area-volume constant) and d_{1j} (height-volume constant) are listed in Table 2. The objective function according to Eq. (34) and $k=1$ is as follows:

$$Minimize\, F = \sum_{i=1}^{n}\sum_{t=1}^{T}(1 - \frac{P_i(t)}{PPC})^2 \tag{38}$$

where F = total power deficit. The decision variables of the objective function are the reservoir releases $R_i(t)$.

Table 1
Reservoir inflow (Q) and reservoir evaporation (Ev) in 60 operating periods.

Year	Q (10^6m^3)					Ev (mm)
	1	2	3	4	5	For 5 years
Month 1	217.4	191.6	210.8	160.8	128.3	158.4
2	220.2	220.2	211	186.2	123.4	77.9
3	250.7	240.4	226.0	205.1	213.7	55.2
4	148.4	199.0	161.3	124.9	136.7	49.9
5	262.6	701.8	354.9	326.9	216.9	64.4
6	344.4	1012.7	943.0	455.4	452.3	80.7
7	1118.3	1969.5	1031.9	714.2	613.8	131.1
8	1120.3	1170.9	762.6	548.4	486.8	165.8
9	738.5	722.8	475.3	340.9	307.1	238.3
10	431.5	463.6	290.9	205.6	181.4	253.3
11	264.8	305.7	218.8	164.3	141.6	259.8
12	208.2	233.9	184.5	135.4	125.8	208.2

4.2. Four-reservoir system operation

This problem was introduced and solved by Chow and Cortes-Rivera (1974) and by Murray and Yakowitz (1979). Bozorg-Haddad et al. (2011) solved this problem using HBMO, considering 220 populations in the HBMO and using 5000 iterations (approximately 1 billion evaluations). Also, this problem is solved by Bozorg-Haddad et al. (2015, 2014a, b) using BBO (Biography-Based Optimization), BA (Bat Algorithm) and WLA (Water Cycle Algorithm) EAs respectively. In order to compare GSA with those proposed by other researchers, the problem is solved in this paper as well. Notice that this problem is a hypothetical example and accordingly, the used data are without units (i.e., they are normalized).

The system consists of four-reservoirs, as shown in Fig. 5(b) and reservoir releases are used for irrigation demand. The objective function is to maximize the sum of benefits from the four reservoirs according to Eq. (34) with $k=0$ (or minimized the minus of total benefits). Each reservoir is operated in 1 year, or 12 operating periods (months). The initial storages equal the ending storages (periodicity constraint). The initial storages are:

$$S_i(1) = \begin{bmatrix} 6 & 6 & 6 & 1 \end{bmatrix} \qquad i = 1, 2, 3, 4 \tag{39}$$

The matrix reservoir connectivity for the four-reservoir problem is:

$$M_{4\times 4} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix} \end{array} \tag{40}$$

where 1, 2, 3 and 4 are the numbers (No.) of the reservoirs, and the numbers 1 and −1 in the matrix show inflow and outflow to the reservoirs, respectively. The inflows $Q_3(t)$ and $Q_4(t)$ are zero and, according to Eq. (40), the releases of Reservoir 2 are the inflows to Reservoir 3 and the releases of the Reservoirs 3 and 1 are the inflows to Reservoir 4 [Fig. 5(b)]. Constraints imposed on reservoir releases are:

$$\begin{bmatrix} 0.005 \\ 0.005 \\ 0.005 \\ 0.005 \end{bmatrix} \le R_i(t) \le \begin{bmatrix} 4 \\ 4.5 \\ 4.5 \\ 8 \end{bmatrix} \quad i = 1, 2, 3, 4;\ t = 1, 2, \ldots, T \tag{41}$$

The penalties C_{max}, C_{min}, and C_{target} equal 40, 40, and 60, respectively. The river inflows, Q, $Smax$, and $Smin$ in the 12 operating periods are listed in Table 3. The benefit functions, b, are listed in Table 4. The objective function according to Eq. (34) is as follows ($K=0$):

$$Maximize\, F = \sum_{i=1}^{n}\sum_{t=1}^{T} a_i(t) \times R_i(t) - \sum_{i=1}^{n}\sum_{t=1}^{T} Pemax_i(t)$$

Table 2
Area-storage coefficients and height-storage coefficients for hydropower production reservoir.

Coefficient	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$
b_1	1.0143	0.0357	-2.9825×10^{-5}	1.4117×10^{-8}	-2.3491×10^{-12}
d_1	864.6173	0.3077	-0.0003	1.5737×10^{-7}	-2.7228×10^{-11}

Table 3
Reservoir inflow (units) and storage (units) constraints in 12 operating periods for the four- reservoir system problem.

Reservoir	*Smax*			*Smin*			Q	
	1	2	3	4	For 4 reservoirs		1	2
Month	1	12	15	8	15	1	0.50	0.40
	2	12	15	8	15	1	1.00	0.70
	3	10	15	8	15	1	2.00	2.00
	4	9	12	8	15	1	3.00	2.00
	5	8	12	8	15	1	3.50	4.00
	6	8	12	8	15	1	2.50	3.50
	7	9	15	8	15	1	2.00	3.00
	8	10	17	8	15	1	1.25	2.50
	9	10	18	8	15	1	1.25	1.30
	10	12	18	8	15	1	0.75	0.75
	11	12	18	8	15	1	1.75	1.75
	12	12	15	8	15	1	1.00	1.00

Table 5
Characteristics of the GSA and GA applied to the hydropower-reservoir problem.

GSA parameters	
G_0	50
α	1
GA parameters	
Mutation rate	0.05
Mutation function	Uniform
Crossover function	0.6
Crossover fraction	Roulette wheel
GSA and GA parameters	
Population	70
Iteration	1000
Objective function evaluation	70,000

$$-\sum_{i=1}^{n}\sum_{t=1}^{T}Pemin_i(t)-\sum_{i=1}^{n}\sum_{t=1}^{T}Petarget_i(t) \qquad (42)$$

The decision variables of the objective function are the reservoir releases $R_i(t)$.

5. Results and discussions

The results of the single-reservoir operation and four-reservoir operation are discussed in the next two sections. Those corresponding to the benchmark mathematical functions are found in the Appendix.

5.1. Results for single reservoir operation (power production)

The optimal operation of single reservoir (Karun4) was solved using the GSA method, and compared with solutions calculated with GA and nonlinear programming (NLP). Also, the results of this study were compared with that of proposed by Bozorg-Haddad et al. (2015, 2014a, b) using BBO, BA, and WLA EAs. The GSA and GA were programmed with the MATLAB (2007). Also, to achieve the global solution, the NLP method was implemented in Lingo 11.0 (obtained after 16 hours of processing time with Intel Core i7-2.93 GHz processor). The parameters of the GSA and GA algorithms are listed in Table 5. Notice that without having the global solution from NLP it would be impossible to assess how accurate are the GSA and GA solutions.

The GSA and GA were initialized with random solutions, thus, there always is a possibility that an individual run may misrepresent their capabilities. Therefore, 10 different runs of GA and GSA were carried out with the same number of functional evaluations

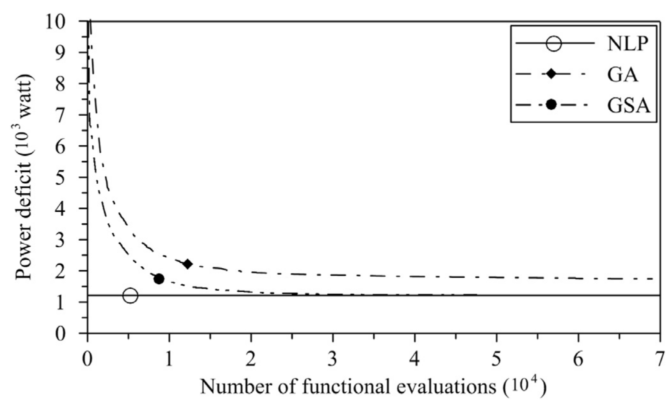


Fig. 6. Convergence paths to a near-optimal solution with 70,000 functional evaluations in hydropower-reservoir problem.

(equal to 70,000). The global solution was 1.213×10^3 W using the NLP method.

The average convergence paths of GSA and GA are presented in Fig. 6. These results establish that the GSA has better and faster convergence than GA towards the optimal solution.

The results of 10 different runs of the GSA and GA are presented in Table 6 and Fig. 7. The coefficient of variation of the GSA is less than the coefficient of variation of the GA and consequently the precision of the GSA exceeds that of the GA (see Table 6). The relative error associated with the average value of the objective function with 70,000 GSA functional evaluations compared to the optimal solution is 0.4%. The relative error equation was calculated according to (GSA result- optimal solution) $\times$ 100/ optimal solution. In summary, Fig. 7 shows that the GSA outperforms the GA in solving this reservoir production problem. The best and the worst

Table 4
Benefit functions in 12 operation periods for four-reservoir system problem.

Month		1	2	3	4	5	6	7	8	9	10	11	12
Reservoir	1	1.1	1.0	1.0	1.2	1.8	2.5	2.2	2.0	1.8	2.2	1.8	1.4
	2	1.4	1.1	1.0	1.0	1.2	1.8	2.5	2.2	2.0	1.8	2.2	1.8
	3	1.0	1.0	1.2	1.8	2.5	2.2	2.0	1.8	2.2	1.8	1.4	1.1
	4	2.6	2.9	3.6	4.4	4.2	4.0	3.8	4.1	3.6	3.1	2.7	2.5

Table 6
Results for 10 different runs of the hydropower-reservoir problem.

No. of run		GSA*(10^3 W)	GA* (10^3 W)	NLP (10^3 W)
	1	1.219	1.672	1.213
	2	1.218	1.549	
	3	1.218	1.864	
	4	1.216	1.752	
	5	1.218	1.986	
	6	1.217	1.752	
	7	1.218	1.931	
	8	1.219	1.569	
	9	1.216	1.841	
	10	1.217	1.534	
	Best	1.216	1.534	
	Worst	1.218	1.986	
Reliability	Average	1.217	1.745	
	Standard deviation	0.0009	0.161	
	Coefficient of variation	0.0007	0.092	

* with 70,000 functional evaluations employing the GSA and the GA

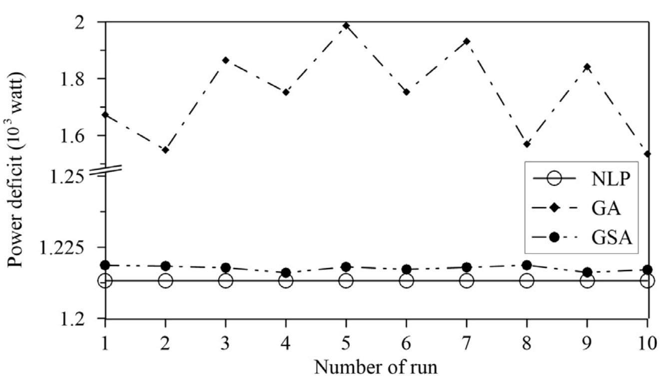


Fig. 7. The results of 10 different runs for the hydropower-reservoir problem.

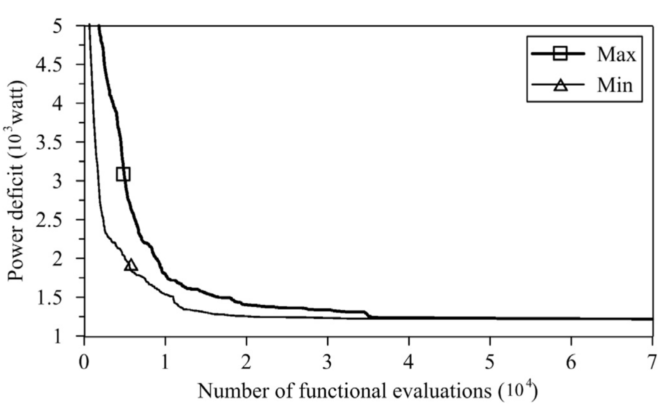


Fig. 8. The best and the worst convergence paths over 10 runs for the hydropower-reservoir problem.

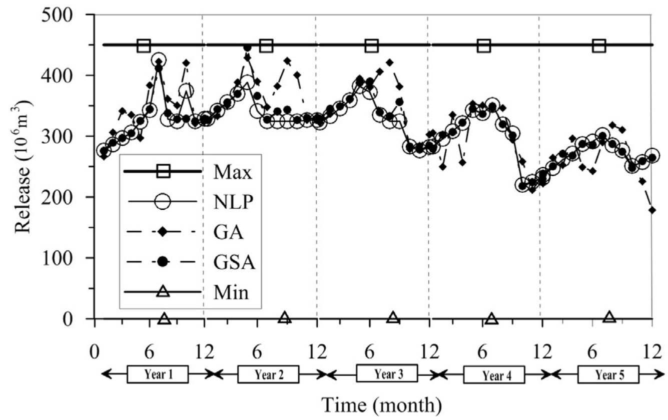


Fig. 9. Monthly reservoir release for the hydropower-reservoir problem.

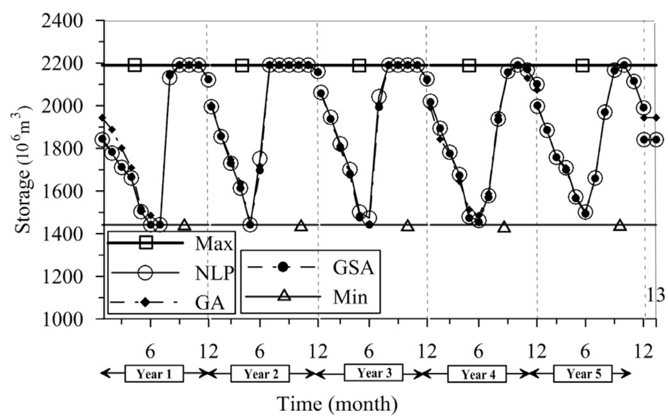


Fig. 10. Monthly storage for the hydropower-reservoir problem.

Table 7
Characteristics of GSA and GA used in four-reservoir system problem.

GSA parameters	
G_0	3
α	5
GA parameters	
Mutation rate	0.06
Mutation function	Uniform
Crossover function	0.7
Crossover fraction	Roulette wheel
GSA and GA parameters	
Population	200
Iteration	2500
Objective function evaluations	500,000

results of 10 runs with 70,000 functional evaluations are plotted in Fig. 8. Also, the worse and the best objective function values are 1.218 and 1.216 (10^3 W), respectively. Notice that the processing time of the GSA algorithm was less than two minutes in each run. A decreasing value of the objective function with increasing number of functional evaluations is evident in Fig. 8.

Monthly reservoir release and storage are presented in Figs. 9 and 10, respectively. These results show that all releases and storage are in the range of feasible solutions. The GSA results are closer to the optimal solution than the GA results. For example, at the beginning and end of the operation period in Fig. 10 the GA results and differ from those of NLP noticeably.

5.2. Results for the four-reservoir system operation

The optimal operation of the four-reservoir was solved using the GSA method, and compared with solutions calculated with GA and LP. The GSA and GA were programmed with MATLAB (2007). The LP method was applied employing Lingo 11.0 and the obtained global solution was 308.2, which is similar to the reported result by Chow and Cortes-Rivera (1974). The quality of the GSA solutions was determined by comparing its results with global solutions. The parameters of GSA and GA are listed in Table 7.

Ten different runs of the GA and the GSA were analyzed and compared using 500,000 functional evaluations. Table 8 lists the results of 10 different runs. The results show that coefficient of variation of the GSA is very low and its reliability is very high. The relative error associated with the average value of the objec-

Table 8
Results for 10 different runs in four-reservoir system problem.

	No. of run	GSA*	GA*	LP
	1	308.5	300.4	308.2
	2	308.7	298.8	
	3	308.6	300.0	
	4	308.3	300.4	
	5	307.9	298.4	
	6	308.1	300.0	
	7	308.1	299.2	
	8	308.5	299.8	
	9	308.5	299.2	
	10	308.7	300.3	
Reliability	Best	308.6	300.4	
	Worst	307.8	298.4	
	Average	308.3	299.6	
	Standard deviation	0.277	0.705	
	Coefficient of variation	0.0008	0.002	

* With 500,000 functional evaluations employing the GSA and the GA

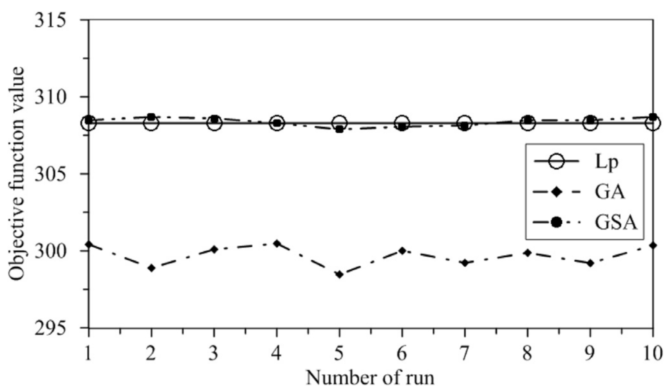


Fig. 11. The results of 10 different runs for the four-reservoir problem.

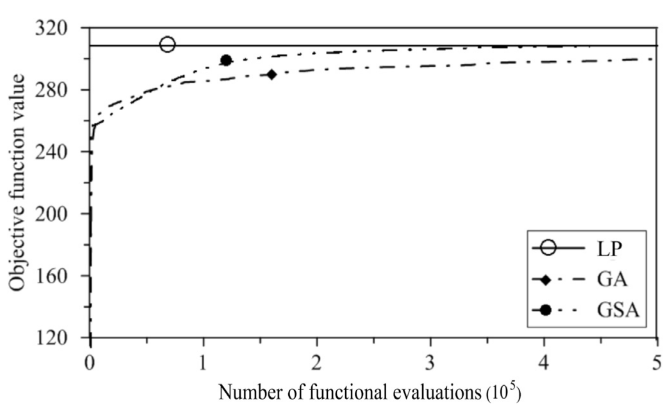


Fig. 12. Convergence paths to a near-optimal solution with number of functional evaluation for the four-reservoir system.

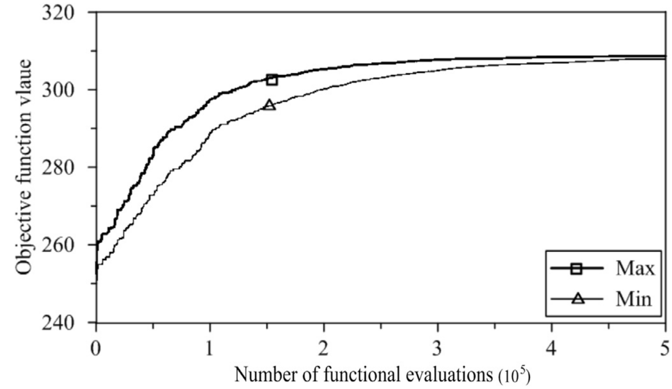


Fig. 13. The best and the worst convergence paths over 10 runs for the four-reservoir problem with the GSA.

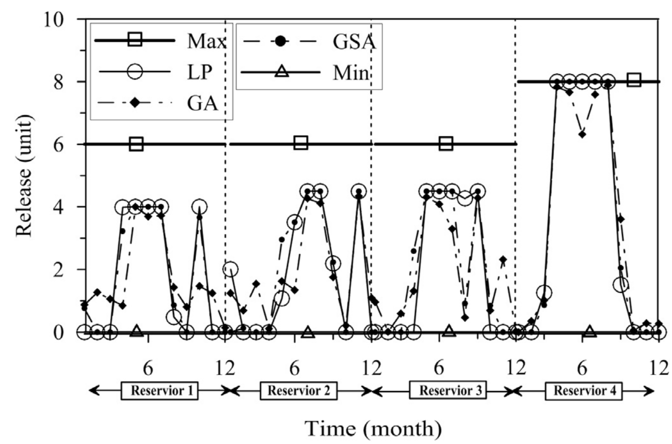


Fig. 14. Monthly release (unit) in the four-reservoir problem.

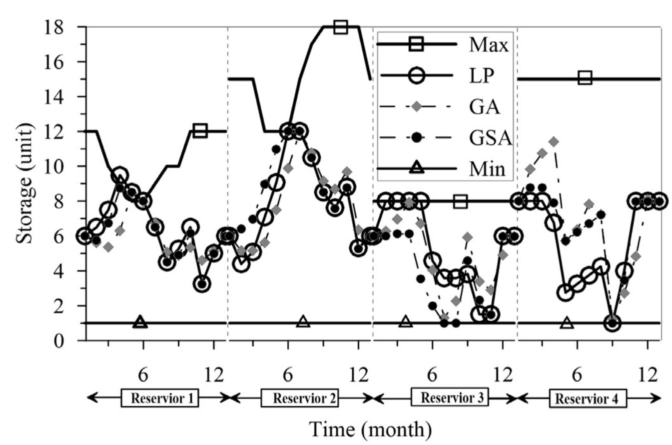


Fig. 15. Monthly storage (unit) in the four-reservoir problem.

tive function with 500,000 functional evaluations for GSA and GA are about 0.4% and 0.7% compared to global solution (the LP result), respectively (see Fig. 11).

The convergence paths of the GSA and the GA are compared with the global solution's in Fig. 12, where it is established the superior performance of the GSA relative to the GA. It is clear that after 500,000 functional evaluations there is a better convergence to the near-optimal solution by the GSA than by the GA.

Results obtained for the best and the worst solution in 10 runs are presented in Fig. 13. Monthly releases and storages for the four reservoirs along with the allowable range of releases and storages are presented in Figs. 14 and 15, respectively. It is noted that all releases and storages are within the range of feasible solutions with no constraint violations.

5.3. Comparing GSA results with the results of BA, BBO, and WCA

In order to evaluate the ability of GSA, single reservoir operation and four-reservoir system operation were solved using GSA. Also, Bozorg-Haddad et al. (2015, 2014a, b) solved these problems with BBO, BA and WLA EAs with the same data and the same function evaluations.

Ten different runs of the GSA and the BBO, BA and WLA were compared in this study. Note that in all cases, 70,000 function eval-

Table 9
Results for 10 different runs of GSA, BA, WLA and BBO in single reservoir operation and four-reservoir system operation.

Number of run	Single reservoir operation				Four-reservoir system operation			
	BA	WLA	BBO	GSA	BA	WLA	BBO	GSA
1	1.233	1.289	1.232	1.219	308.20	306.83	308.00	308.50
2	1.235	1.269	1.239	1.218	307.12	302.40	308.02	308.70
3	1.254	1.287	1.227	1.218	307.41	303.65	308.12	308.60
4	1.236	1.260	1.235	1.216	307.93	303.60	307.56	308.30
5	1.237	1.289	1.223	1.218	308.09	302.38	307.11	307.90
6	1.250	1.285	1.223	1.217	307.95	306.01	307.88	308.10
7	1.237	1.281	1.227	1.218	308.09	304.05	307.57	308.10
8	1.234	1.279	1.223	1.219	308.03	306.75	308.08	308.50
9	1.235	1.286	1.229	1.216	307.62	306.63	308.00	308.50
10	1.241	1.262	1.225	1.217	308.02	306.92	306.55	308.70
Global solution		1.213				308.20		
Average	1.254	1.279	1.228	1.217	307.12	304.92	307.69	308.30
Relative error%	3.41	5.44	1.23	0.33	0.350	1.064	0.165	0.032
Worst	1.233	1.289	1.223	1.216	308.20	306.92	308.12	308.70
Best	1.239	1.260	1.239	1.218	307.84	302.38	306.55	308.10
Standard deviation	0.007	0.010	0.005	0.001	0.350	1.887	0.511	0.277
Coefficient of variation	0.006	0.008	0.004	0.001	0.001	0.006	0.002	0.001

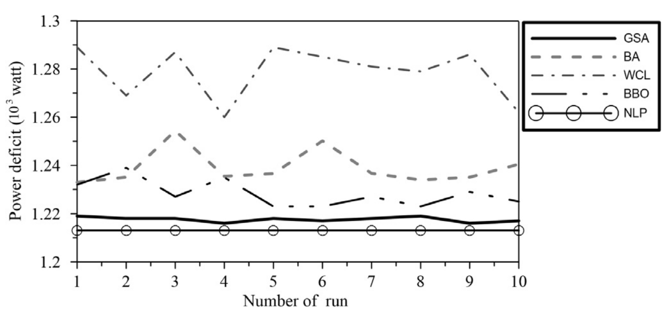


Fig. 16. The results of 10 different runs problem of GSA, BA, WCA and BBO for the single reservoir operation.

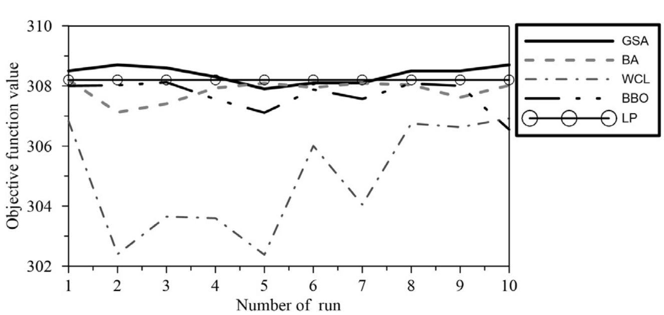


Fig. 17. The results of 10 different runs of GSA, BA, WCA and BBO for the four-reservoirs system operation.

uation in single reservoir operation and 500,000 functional evaluations in four-reservoirs system operation were considered. Table 9 lists the results of the 10 different runs. The results show that, the coefficient of variation and the standard deviation associated with GSA results are significantly lower than other EAs. The average value of the 10-objective function in GSA are more close to global solutions (see Figs. 16 and 17). Also, the relative error associated with the average value of the objective function for GSA, BBO, BA and WLA were calculated and listed in Table 9. The computed relative errors associated with GSA was considerably lower than other EAs.

The convergence paths of the GSA, BBO, BA and WLA were compared with the global solutions in Figs. 18 and 19. The results show that GSA has better and faster convergence than other algorithms towards the optimal solution where it is established the superior

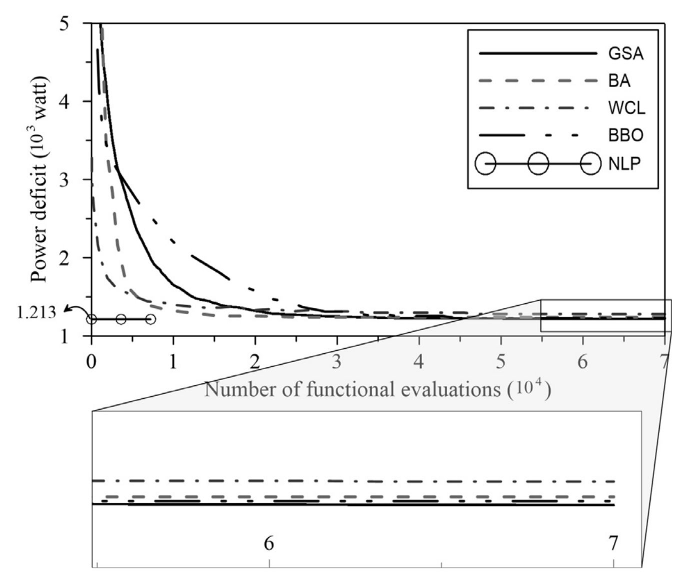


Fig. 18. Convergence paths of GSA, BA, WCA and BBO to a NLP solution with number of functional evaluations for the single reservoir operation.

performance of the GSA relative to these EAs (BBO, BA and WLA). In summary, the results of the GSA demonstrate its ability and efficiency for solving water-resource optimization problems.

6. Concluding remarks

Evolutionary optimization algorithms have been widely used to solve complex water management problems. The gravity Search Algorithm, GSA, which is based on the Newton's law of gravitation, is one type of such evolutionary optimization algorithm. The GSA's performance was evaluated in this paper by minimizing three benchmark functions, i.e. Bukin6, Rosenbrock, and Sphere and by solving a hydropower optimization problem, and a four-reservoir optimization problem. Consequently, the applicability and scalability of GSA were evaluated by using different type of objective functions such as LP and NLP and also with different scale or type of problems such as single reservoir operation and multi-reservoir operation problems. The GSA was compared to the Genetic Algorithm (GA), a well-known and reliable EA, for solving these sets of problems. The results indicated that the GSA could

Table 10
Specification of the benchmark functions.

Function	Sphere	Rosenbrock	Bukin6
Equation	$\sum_{1=1}^{n} x_i^2$	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$100\sqrt{\lvert x_2 - 0.01x_1^2 \rvert} + 0.01\lvert x_1 + 10 \rvert$
Global minimum	$f(x^*) = 0$ $x^* = (0, \ldots, 0)$	$f(x^*) = 0$ $x^* = (1, \ldots, 1)$	$f(x^*) = 0$ $x^* = (-10, 1)$
Variable range	$-5.12 \le x_i \le 5.12$	$-2.048 \le x_i \le 2.048$	$-15 \le x_1 \le -5, -15 \le x_2 \le -5$

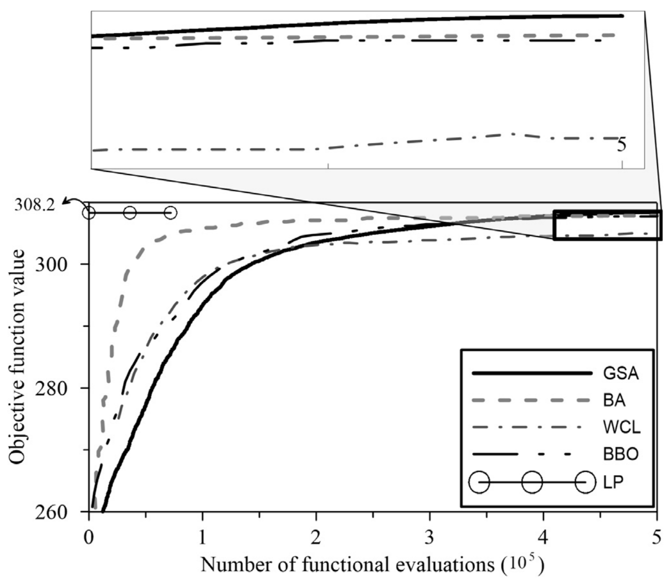


Fig. 19. Convergence paths of GSA, BA, WCA and BBO to a LP solution with number of functional evaluations for the four- reservoir system.

find solutions more rapidly and with better accuracy than the GA. The relative errors of the average value of the GSA objective function associated with the hydro-power reservoir optimization (the Karun4 reservoir) and four-reservoir system compared to the optimal solution were equal to 0.4% and 0.3% respectively. The relative errors of the GA were 30% and 3% of the optimal solutions for the two same reservoir problems, respectively. The standard deviations of the GSA results were lower than those of the GA in 10 runs of the operations of the single and multi-reservoir problems. The standard deviations of the GSA were 0.0009 and 0.277 for the Karon4 reservoir and the four-reservoir operations system, respectively. Concerning the GA results the standard deviations of its solutions from 10 runs were 0.161 and 0.705 for the single reservoir and four-reservoir problems, respectively. Consequently the reliability of the GSA exceeded the GA's. Approximately similar numbers of populations, iterations, and objective function evaluations were applied to obtain the associated parameters of the GSA and the GA. However, the results indicated that the tuning of the GSA parameters was simpler and its run time was faster than the GA's. This is another advantage of the GSA over the GA. In summary, the results of the GSA demonstrate its applicability, scalability, and efficiency for solving water-resource optimization problems.

Appendix. Verifying the GSA with benchmark functions

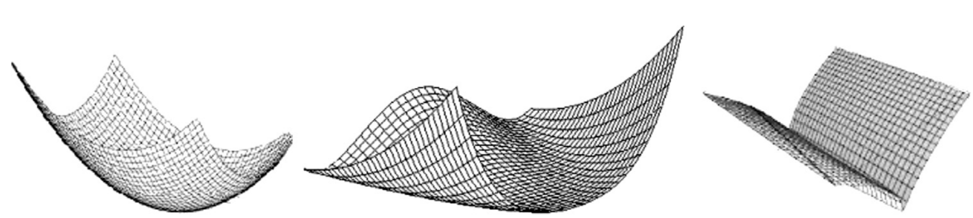
The performance of the GSA algorithm was evaluated with three standard benchmark functions. These benchmark functions are the Sphere, Rosenbrock and Bukin6 functions that have strikingly different geometries but have a minimum equal to zero in all three cases. Details of the three functions are presented in Fig. 20 and Table 10.

The GSA was compared with the GA [GSA and GA were programmed with MATLAB (2007)] by minimizing the three benchmark functions applying 9001 functional evaluations in each of 10 runs used to approximate the global minimum. Notice that each of the 10 runs was reported to ensure the results are not fortuitous. From these evaluations the worst, the best, and the average solutions from the 10 runs were calculated.

The results and statistical measures derived from the 10 different runs are listed in Fig. 21 and Table 11. Notice that Fig. 21 displays the unique solution (equal to zero) obtained with nonlinear programming (NLP) implemented by using LINGO 11.0. The very low coefficient of variation from the 10 runs is an indication of the convergence to a close region enveloping the global minimum. The average values of the GSA objective functions for 10 different runs with 9001 functional evaluations each are closer to the global minima than those obtained with the GA. Since for single objective algorithms, reliability is represented by the average and standard deviation (Marchi et al., 2014), it can be said the reliability of GSA was more than GA. On the other hand, the increasing reliability is associated with decreasing the standard division and the average EAs' results in severa runs are close to the global solution.

Fig. 22 presents the convergence paths for the GA and GSA. It is seen in Fig. 22 that the GSA converges close to the global minima after 9001 functional evaluations. The convergence paths of the GSA to the best and the worst solutions over the 10 runs are presented in Fig. 23.

The previous results confirm the excellent performance of the GSA method in solving the chosen nonlinear functions, in which it fared better than the GA.



(a) (b) (c)

Fig. 20. Benchmark functions: (a) Sphere, (b) Rosenbrock, and (c) Bukin6 (The global minimum equals zero in each function).

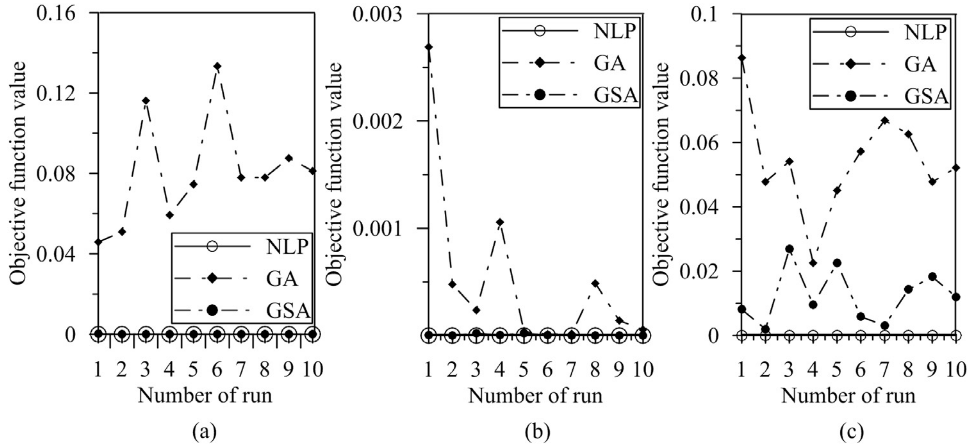


Fig. 21. The results of 10 different runs for (a) Sphere, (b) Rosenbrock, and (c) Bukin6 functions.

Table 11
Minimization results for 10 different runs using benchmark functions.

Problem			Sphere		Rosenbrock		Bukin6	
			GA	GSA*	GA	GSA*	GA	GSA*
		1	0.04	0.00016	0.00269	0.0000013	0.08	0.008
		2	0.05	0.00009	0.00047	0.0000004	0.04	0.001
		3	0.11	0.00007	0.00023	0.0000188	0.05	0.026
	No. of run	4	0.05	0.00006	0.00105	0.0000045	0.02	0.009
		5	0.07	0.00006	0.00003	0.0000206	0.04	0.022
		6	0.13	0.00009	0.00001	0.0000024	0.05	0.005
		7	0.07	0.00006	0.00001	0.0000007	0.06	0.003
		8	0.07	0.00005	0.00048	0.0000006	0.06	0.014
		9	0.08	0.00005	0.00013	0.0000010	0.04	0.018
		10	0.08	0.00009	0.00005	0.0000002	0.05	0.012
	Best		0.04	0.00005	0.00001	0.0000002	0.02	0.001
	Worst		0.13	0.00016	0.00269	0.0000206	0.08	0.026
Reliability	Average		0.08	0.00008	0.00052	0.0000050	0.05	0.012
	Standard deviation		0.02	0.00003	0.000829	0.0000078	0.01	0.008
	Coefficient of variation		0.33	0.42000	1.595149	1.5340490	0.30	0.674

* with 9001 functional evaluations employing the GSA and GA algorithm (10 population, 1000 iteration and 1 elite selection)

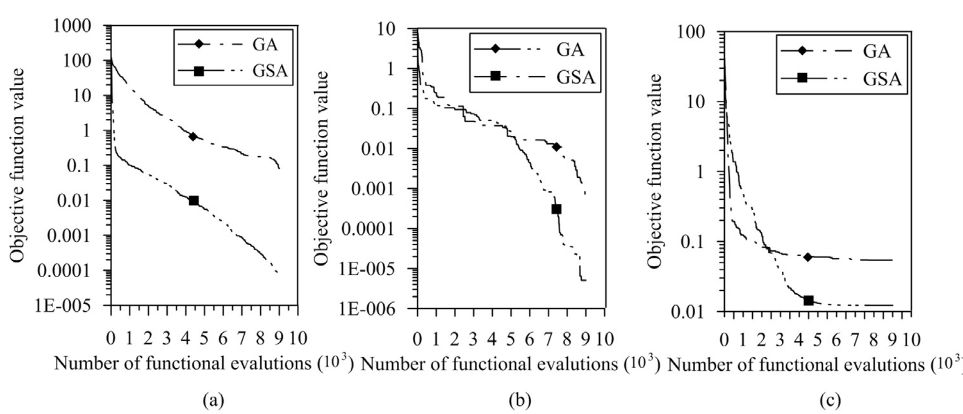


Fig. 22. Convergence paths to a near-optimal solution with number of functional evaluations for (a) Sphere, (b) Rosenbrock, and (c) Bukin6 functions.

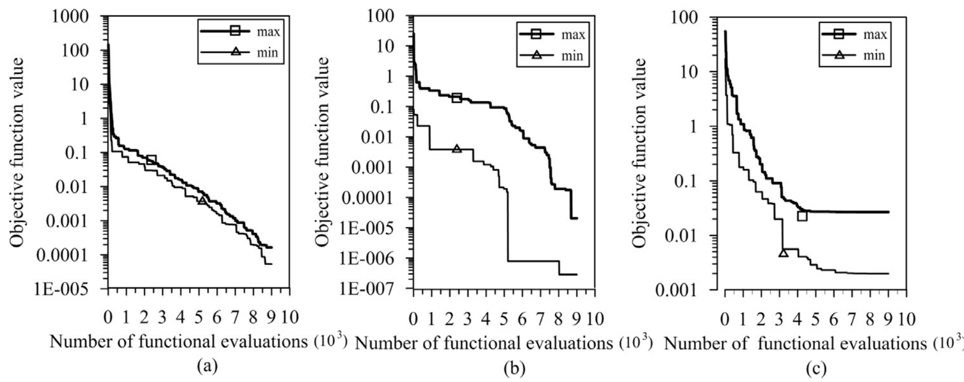


Fig. 23. The best and the worst solutions over 10 runs for the (a) Sphere, (b) Rosenbrock, and (c) Bukin6 functions.

References

Ahmadi, M., Bozorg-Haddad, O., Mariño, M.A., 2014. Extraction of flexible multi--objective real-time reservoir operation rules. Water Resour. Manage. 28 (1), 131–147.

Ahmadianfar, I., Adib, A., Salarijazi, M., 2015. Optimizing multireservoir operation: hybrid of bat algorithm and differential evolution. J. Water Resour. Plann. Manag. (ASCE), 05015010 http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000606.

Ashofteh, P.S., Bozorg-Haddad, O., Loáiciga, H.A., 2015. Evaluation of climatic-change impacts on multi-objective reservoir operation with multiobjective genetic programming. J. Water Resour. Plann. Manag. (ASCE) 141 (11). http://dx.doi.org/10.1061/(ASCE)WR. 1943-5452.0000540.

Ashofteh, P.-S., Bozorg-Haddad, O., Mariño, M.A., 2013. Climate change impact on reservoir performance indices in agricultural water supply. J. Irrig. Drain. Eng. (ASCE) 139 (2), 85–97. http://dx.doi.org/10.1061/(ASCE)IR.1943-4774.0000496.

Bolouri-Yazdeli, Y., Bozorg-Haddad, O., Fallah-Mehdipour, E., Mariño, M.A., 2014. Evaluation of real-time operation rules in reservoir systems operation. Water Resour. Manage. 28 (3), 715–729.

Bozorg-Haddad, O., Afshar, A., Mariño, M.A., 2011. Multi-reservoir optimization in discrete and continuous domains. Proc. Inst. Civ. Eng. Water Manage. 164 (2), 57–72.

Bozorg-Haddad, O., Hosseini-Moghari, S., Loáiciga, H., 2015. Biogeography-based optimization algorithm for optimal operation of reservoir systems. J. Water Resour. Plann. Manage, 04015034 http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000558.

Bozorg-Haddad, O., Karimirad, I., Seifollahi-Aghmiuni, S., Loáiciga, H., 2014a. Development and application of the bat algorithm for optimizing the operation of reservoir systems. J. Water Resour. Plann. Manage., 04014097 http://dx.doi.org/10.1061/(ASCE)WR.1943 5452.0000498.

Bozorg-Haddad, O., Mirmomeni, M., Marino, M.A., 2010. Optimal design of stepped spillways using the HBMO algorithm. Civil Eng. Env. Syst. 27 (1), 81–94.

Celeste, A.B., Billib, M., 2009. Evaluation of stochastic reservoir operation optimization models. Adv. Water Res. 32 (9), 1429–1443.

Chen, Z., Yuan, X., Tian, H., Ji, B., 2014. Improved gravitational search algorithm for parameter identification of water turbine regulation system. Energy Convers. Manage. 78, 306–315.

Chow, V.T., Cortes-Rivera, G., 1974. Application of DDDP in Water Resources Planning. University of Illinois at Urbana-Champaign, Water Resources Center.

Côté, P., Leconte, R., 2015. Comparison of stochastic optimization algorithms for hydropower reservoir operation with ensemble streamflow prediction. J. Water Resour. Plann. Manage., 04015046 http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000575.

Ficchì, A., Raso, L., Dorchies, D., Pianosi, F., Malaterre, P., Van Overloop, P., Jay-Allemand, M., 2015. Optimal operation of the multireservoir system in the seine river basin using deterministic and ensemble forecasts. J. Water Resour. Plann. Manage., 05015005 http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0000571.

Ghalambaz, M., Noghrehabadi, A.R., Behrang, M.A., Assareh, E., Ghanbarzadeh, A., Hedayat, N., 2011. A Hybrid Neural Network and Gravitational Search Algorithm (HNNGSA) Method to Solve well known Wessinger's Equation. World Academy of Science Engineering and Technology 49 (51), 803–807.

Haddad, O., Moravej, M., Loáiciga, H., 2014b. Application of the water cycle algorithm to the optimal operation of reservoir systems. J. Irrig. Drain Eng 141 (8).

Jadidi, Z., Muthukkumarasamy, V., Sithirasenan, E., Sheikhan, M., 2013. Flow-based anomaly detection using neural network optimized with GSA algorithm. In: Distributed Computing Systems Workshops (ICDCSW), IEEE 33rd International Conference, pp. 76–81.

Lingo 11.0 [Computer Software], 2016. Chicago, Lindo.

Marchi, A., Dandy, G.C., Wilkins, A., Rohrlach, H., 2014. Methodology for comparing evolutionary algorithms for optimization of water distribution systems". J.Water Resour. Plan. Manag. 140 (1), 22e31.

MATLAB version 7.11.0, 2007. Computer Software). The Mathworks Inc, Natick, Massachusetts.

Mousavi, S.J., Mahdizadeh, K., Afshar, A., 2004. A stochastic dynamic programming model with fuzzy storage states for reservoir operations. Adv. Water Resour. 27 (11), 1105–1110 Elsevier.

Murray, D.M., Yakowitz, S.J., 1979. Constrained differential dynamic programming and its application to multi-reservoir control. Water Resour. Res. Wiley 15 (5), 1017–1027.

Oliveira, R., Loucks, D.P., 1997. Operating rules for multi reservoir systems. Water Resour. Res 33 (4), 839–852.

Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2009. GSA: a gravitational search algorithm. Inf. Sci. Elsevier 179 (13), 2232–2248.

Tilmant, A., Kelman, R., 2007. A stochastic approach to analyze trade-offs and risks associated with large-scale water resources systems. Water Resour. Res. 43, W06425. http://dx.doi.org/10.1029/2006WR005094.

Vesterstrom, J., Thomsen, R., 2004. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. IEEE Trans. Evol. Comput. 2, 1980–1987.

Wang, K., Chang, L.C., Chang, F.J., 2011. Multi-tier interactive genetic algorithms for the optimization of long-term reservoir operation. Adv. Water Resour. Elsevier 34 (10), 1343–1351.

Wardlaw, R., Sharif, M., 1999. Evaluation of genetic algorithms for optimal reservoir system operation. J. Water Resour. Plann. Manag. ASCE 125 (1), 25–33.

Wehrens, R., Buydens, L.M., 2006. Classical and nonclassical optimization methods. Encyclopedia Analy. Chem. 9678–9689.

Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. Trans. Evol. Comput. IEEE 3 (2), 82–102.