RESERVOIR OPERATION OPTIMIZATION USING CBO, ECBO AND VPS ALGORITHMS

M.H. Rabiei[1], M.T. Aalami[1] and S. Talatahari[1,2,*,†]
[1]Department of Civil Engineering, University of Tabriz, Tabriz, Iran,
[2]Engineering Faculty, Near East University, 99138 Nicosia, North Cyprus, Mersin 10, Turkey

ABSTRACT

This paper utilizes the Colliding Bodies of Optimization (CBO), Enhanced Colliding Bodies of Optimization (ECBO) and Vibrating Particles System (VPS) algorithms to optimize the reservoir system operation. CBO is based on physics equations governing the one-dimensional collisions between bodies, with each agent solution being considered as an object or body with mass and ECBO utilizes memory to save some historically best solutions and uses a random procedure to escape from local optima. VPS is based on simulating free vibration of single degree of freedom systems with viscous damping. To evaluate the performance of these three recent population-based meta-heuristic algorithms, they are applied to one of the most complex and challenging issues related to water resource management, called reservoir operation optimization problems. Hypothetical 4 and 10-reservoir systems are studied to demonstrate the effectiveness and robustness of the algorithms. The aim is on discovering the optimum mix of releases, which will lead to maximum benefit generation throughout the system. Comparative results show the successful performance of the VPS algorithm in comparison to the CBO and its enhanced version.

1. INTRODUCTION

Reservoirs provide the main water resources in many basins. Optimal operation of

---

*Corresponding author: Department of Civil Engineering, University of Tabriz, Tabriz, Iran
†E-mail address: siamak.talat@gmail.com (S. Talatahari)

reservoirs is necessary in regional water management especially in case of limited water resources. Reservoirs Operation Optimization (ROO) problems typically are difficult to solve because of the involvement of a large number of decision variables and constraints. Thus, solving these problems need a powerful technique.

Two different methods have already been applied for solving the reservoir operation optimization problem, classical methods and evolutionary algorithms (EAs). Linear programming (LP), dynamic programming (DP), stochastic dynamic programming (SDP) and non-linear programming (NLP) are settled in classical methods category. Various researchers have applied these classical methods to support the decision-making process of water reservoir operations [1-4]. Almost all these classical optimization methods have limitations in solving ROO problems. For example, LP only can solve optimization problems with linear objective function and constraints, DP and SDP suffer from curse of dimensionality and state-space discretization and NLP may trap in a local optimum especially in non-convex optimization problems [5].

EAs have been widely used in several fields of water resources system issues such as reservoir operation. Wardlaw and Sharif [6] optimized this problem using genetic algorithm (GA) and reported that algorithm with real value coding performs significantly faster than the one that employs binary coding. According to Cai et al. [7], evolutionary methods have been applied to solve new large-scale nonlinear reservoir management models. They presented a combined GA and linear programming strategy for solving large nonlinear problems that are difficult, if not impossible, using currently available NLP solvers. GA models were also successfully applied by Chen [8]. Ahmed and Sarma [9] presented a genetic algorithm model to find the optimal operating policy of a multipurpose reservoir, located on the river Pagladia, a major tributary of the river Brahmaputra.

Kumar and Reddy [10] made comparison between the performances of ant colony optimization (ACO) and GA in the operation of the Hirakud reservoir in India with agricultural, hydropower, and flood control functions. Their results indicate the superiority of ACO over GA in accuracy and computational speed terms. GA was used to derive optimal operation policies of Pechiparai reservoir in Tamil Nadu, India by Jothiprakash and Shanthi [11]. Adding pheromone re-initiations (PRIs) and partial path replacement (PPR) to ant colony optimization (ACO), Jalali et al. [12] optimized single and multi-reservoir system operation. Zahraie and Hosseini [13] derived operation rules of Zayandeh-Rud River reservoir in Iran. They used GA to determine optimal operation solution and derived the classic and fuzzy regressions operation rules. Bozorg-Haddad et al. [14] studied the capability of honey bee mating optimization (HBMO) algorithm in solving four-reservoir and ten-reservoir operation system problems, in both continuous and discrete domains. Their results indicated that the HBMO algorithm is able to solve such large scale optimization problems. Dariane and Sarani [15] employed the Intelligent Water Drops (IWD) algorithm and the ACO in Dez reservoir operation problem in Iran. Comparison of the results shows that while the IWD algorithm finds relatively better solutions and it is able to overcome the computational time consumption deficiencies inherited in the ACO methods, which is very important in large scale problems with too many decision variables problems where running time becomes a limiting factor for optimization model applications. Bozorg-Haddad et al. [16] applied the bat algorithm (BA) to optimize single and four-

reservoir systems. Their results indicate that the high efficiency of BA in hydropower operation and large scale optimization problems.

Single and four-reservoir system operation problems were solved using the water cycle algorithm (WCA) in [17] and the superiority of WCA over GA was proved. Hosseini-Moghari et al. [18] compared the results of imperialist competitive algorithm (ICA), cuckoo optimization algorithm (COA) and GA with NLP in a single reservoir and four-reservoir optimization problem and the COA was reported as the most accurate method. Bozorg-Haddad et al. [19] applied biogeography-based optimization (BBO) Algorithm to single and four-reservoir systems and the results shows the superiority of BBO over GA and WCA in term of accuracy. Asgari et al. [20] employed weed optimization algorithm (WOA) to solve four-reservoir system problem and reported that the WOA produces better results than those of GA. The efficacy of gravity search algorithm (GSA) for solving ROO problems is also investigated in [21]. The results of the GSA demonstrate its applicability, scalability, and efficiency for solving water-resource optimization problems. Moravej and Moghari [22] applied the interior search algorithm (ISA) to solve one-, four- and ten- reservoirs system operation problems. They compared results of the ISA with those of NLP, GA and other meta-heuristic algorithms. Considering the results, it was stated that the ISA is a powerful tool to optimize complex large scale reservoir system operation problems.

In this study, the Colliding Bodies Optimization (CBO) developed by Kaveh and Mahdavi [23], the Enhanced Colliding Bodies Optimization (ECBO) introduced by Kaveh and Ilchi Ghazaan [24], as well as Vibrating Particles System (VPS) developed by Kaveh and Ilchi Ghazaan [25] are used to determine optimal operation of hypothetical reservoir problems. Considering the fact that the CBO, ECBO and VPS have not been used to solve water resources management so far and their high capability as reported in [26, 27], a survey on application of these algorithms in multi-reservoir system operation problem seems necessary. Thus, this paper deals with the application of these new algorithms in reservoir system operation problems. To do so, two reservoir systems were considered (four-reservoir and ten-reservoir systems) and the results of other well-studied algorithms were compared with them.

The rest of this paper is organized as follows: CBO, ECBO and VPS are described in the next three sections. In Section 5, modeling of reservoir operation is presented. Results and analysis of proposed algorithms application in optimization of a reservoir operation problem are given in Section 6 and finally, the achievements of the paper are summarized in the conclusion section.

2. COLLIDING BODIES OPTIMIZATION ALGORITHM

The collision is a natural occurrence and the CBO algorithm was developed based on this phenomenon by Kaveh and Mahdavi [23]. In this technique, one object collides with other object and they move towards a minimum energy level. Each CB has initial randomly position in the search space:

$$x_i^0 = x_{min} + rand.\left(x_{max} - x_{min}\right) \quad i = 1, 2, \ldots, n \tag{1}$$

where x_i^0 is the initial solution vector of the ith CB. x_{min} and x_{max} are the minimum and the maximum allowable variables vectors; and *rand* is a random vector with each component being in the interval [0,1]. Also each CB has a specified mass defined base on its objective function value, as:

$$m_k = \frac{1/fit(k)}{\sum_{i=1}^{2n} 1/fit(i)} \quad i = 1, 2, \ldots, n \tag{2}$$

where, *fit*(i) represents the objective function value of the ith CB; n is the population size. Also, for maximizing the objective function, the term $1/fit(i)$ is replaced by $fit(i)$.

The arrangement of the CBs objective function values is performed in ascending order. The sorted CBs are equally divided into two groups: (i) stationary, (ii) moving. The lower half of the CBs (stationary CBs) are good agents which are stationary and the upper half of CBs (moving CBs) move toward the lower half. The velocity of these CBs before collision (v_i) is:

$$v_i = 0 \quad i = 1, 2, \ldots, \frac{n}{2} \tag{3}$$

$$v_i = x_{i-n/2} - x_i \quad i = \frac{n}{2} + 1, \ldots, n \tag{4}$$

where x_i is the position vectors of the ith CB.

Then moving CBs collide to stationary CBs to improve their positions and push stationary CBs towards better positions. After collision, the velocity of stationary and moving CBs (v_i') are evaluated by:

$$v_i' = \frac{(m_{i+n} - \varepsilon m_{i+n}) v_{i+n}}{m_i + m_{i-n}} \quad i = 1, 2, \ldots, n \tag{5}$$

$$v_i' = \frac{(m_i - \varepsilon m_{i-n}) v_i}{m_i + m_{i-n}} \quad i = n+1, \ldots, 2n \tag{6}$$

where m is the mass of the ith CB, ε is the coefficient of restitution (COR) and for most of the real objects, its value is between 0 and 1. It is used to control of the exploration and exploitation rate and defined as:

$$\varepsilon = 1 - \frac{iter}{iter_{max}} \tag{7}$$

where, *iter* is the current iteration number and $iter_{max}$ is the maximum number of iterations.

New positions of each group are obtained using the generated velocities after the collision:

$$x_i^{new} = x_i + rand \circ v_i^{'} \quad i = 1, 2, \ldots, \frac{n}{2} \tag{8}$$

$$x_i^{new} = x_{i-n/2} + rand \circ v_i^{'} \quad i = \frac{n}{2} + 1, \ldots, n \tag{9}$$

where x_i^{new} and $v_i^{'}$ are the new position and the velocity after the collision of the ith CB, respectively; *rand* is a random vector uniformly distributed in the range [-1, 1] and the sign "∘" denotes an element-by-element multiplication. The optimization process is repeated until the termination criterion, specified as the maximum number of iterations, is satisfied.

3. ENHANCED COLLIDING BODIES OPTIMIZATION ALGORITHM

ECBO was developed by Kaveh and Ilchi Ghazan [24] to improve the performance of the CBO and reduce the computational cost. In this enhanced version, a memory is defined to save a number of historically best CBs and also some components of CBs are changed randomly to escape from local optima. The main steps of the ECBO are given as follows:

Step 1. Initialization

The initial positions of all the CBs are created randomly in an m-dimensional search space based on Eq. (1).

Step 2. *Mass allocation*

The value of mass for each CB is allocated according to Eq. (2).

Step 3. *Storing CM*

Colliding memory (CM) is defined to save a number of historically best CB vectors and their related mass and objective function values. In this study the size of the CM is taken as n/4. In this step, the solution vectors that are saved in CM are added to the population, and the same numbers of current worst CBs are deleted. At last, CBs are sorted according to their objective function values.

Step 4. *Dividing*

CBs are divided into two equal groups to select the pairs for collision: (i) stationary group, (ii) moving group.

Step 5. *Calculating velocities*

The velocities of stationary and moving bodies before collision are obtained using Eqs. (3) and (4), respectively. The moving group move toward the stationary group and their velocities after collision are calculated by Eqs. (5) and (6), respectively.

Step 6. *Updating CBs*

The new position of each CB is evaluated base on Eqs. (8) and (9).

Step 7. *Regeneration*

In each iteration, a parameter like *Pro* within (0, 1) is introduced and it is specified whether a component of each CB must be changed or not. For each CB, *Pro* is compared with *rand* which is a random number uniformly distributed within (0, 1). If *rand* < *Pro*, one dimension of the ith CB is selected randomly and its value is regenerated base on:

$$x_{ij} = x_{j,\min} + rand\left(x_{j,\max} - x_{j,\min}\right) \tag{10}$$

where x_{ij} is the jth variable of the ith CB. $x_{j,\max}$ and $x_{j,\min}$ are the lower and upper bounds of the jth variable. In order to protect the structures of CBs, only one dimension is changed. In this paper, the value of *Pro* is set to 0.3.

Step 8. *Terminating criterion check*

Repeat step 2 to step 7 until a terminating criterion is satisfied.

4. VIBRATING PARTICLES SYSTEM OPTIMIZATION ALGORITHM

The VPS is a population-based meta-heuristic method that simulates a free vibration of single degree of freedom systems with viscous damping (Kaveh and Ilchi Ghazaan [25]). The VPS involves a number of particles consisting of the solutions of the problem. The initial positions of particles are created randomly in an n-dimensional search space. The solution candidates gradually approach to their equilibrium positions which are achieved from current population and historically best position in order to have a proper balance between diversification and intensification.

For each particle, three equilibrium positions with different weights are defined, and during each generation, the particle position is updated by learning from them: (i) the historically best position of the entire population (HB), (ii) a good particle (GP), and (iii) a bad particle (BP). In order to select the GP and BP for each candidate solution, the current population is sorted according to their objective function values in an increasing order, and then GP and BP are chosen randomly from the first and second half, respectively.

A descending function based on the number of iterations is proposed in VPS to model the effect of the damping level in the vibration.

$$D = \left(\frac{iter}{iter_{\max}}\right)^{-\alpha} \tag{11}$$

where *iter* is the current iteration number and *iter$_{max}$* is the total number of iterations for the optimization process. α is a constant.

According to the above concepts, the update rules in the VPS are given by:

$$x_i^j = w_1.\left[D.A.rand1 + HB^j\right] + w_2.\left[D.A.rand2 + GP^j\right] + w_3.\left[D.A.rand3 + BP^j\right] \tag{12}$$

$$A = \left[w_1.\left(HB^j - x_i^j\right)\right] + \left[w_2.\left(GP^j - x_i^j\right)\right] + \left[w_3.\left(BP^j - x_i^j\right)\right] \tag{13}$$

$$w_1 + w_2 + w_3 = 1 \tag{14}$$

where x_i^j is the jth variable of the particle i. w_1, w_2, and w_3 are three parameters to measure the relative importance of HB, GP and BP, respectively. *rand1*, *rand2* and *rand3* are random numbers uniformly distributed in the range of [0, 1].

In order to have a fast convergence in the VPS, the effect of BP is just sometimes considered in updating the position formula. Therefore, for each particle, a parameter like p within (0, 1) is defined, and it is compared with rand (a random number uniformly distributed in the range of [0, 1]) and if $p < rand$, then $w_3 = 0$ and $w_2 = 1 - w_1$.

5. MULTI-RESERVOIR OPERATION OPTIMIZATION MODELS

The four-reservoir problem was introduced and solved for continuous decision variables by Chow and Cortes-Rivera [28]. Constrained differential dynamic programming method was used by Murray and Yakowitz [29] to solve this problem. The system consists of four reservoirs in the form of series and parallel, as shown in Fig. 1(a) and reservoir releases are used to generate hydropower and to satisfy irrigation water demand. Hydropower and irrigation benefits are quantified by linear functions of the discharge.

The ten-reservoir problem was formulated and introduced by Murray and Yakowitz [29]. The schematic of this problem is shown in Fig. 1(b). The releases from the upstream reservoirs are passed on to the downstream reservoirs, and a reservoir may receive supplies from one or more upstream reservoirs. The releases from the reservoirs are used for generating hydropower. The benefit function is a linear function of release and is based upon the numerical values provided by Murray and Yakowitz [29].
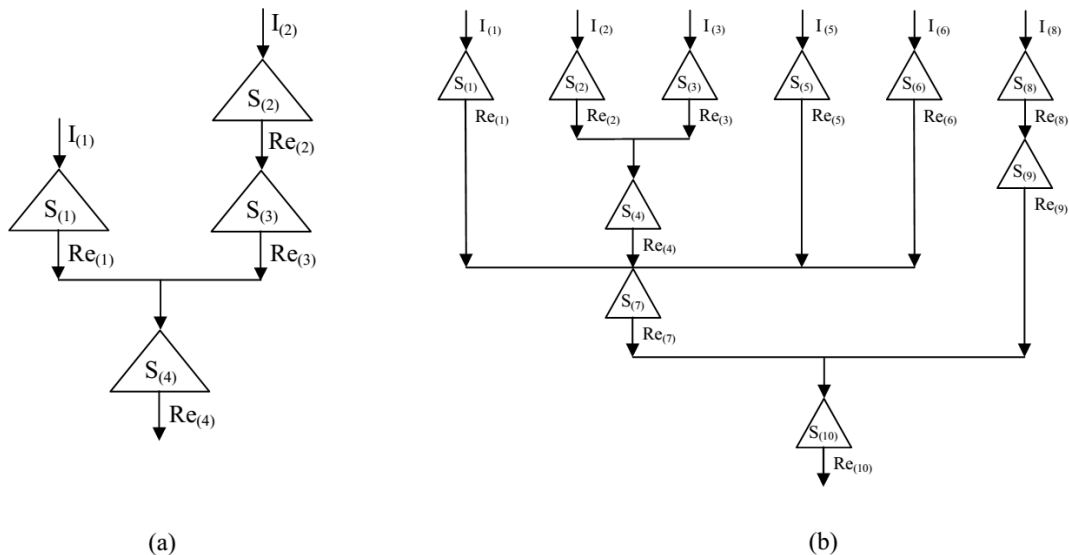

(a)   (b)
Figure 1. Schematic representation of the (a) four-reservoir and (b) ten-reservoir problem

To compare the capability of the CBO, ECBO and VPS algorithms in solving ROO problems, these benchmark problems of multi-reservoir operation are considered. These problems offer an opportunity to test the performance of the proposed algorithms against

known global optima obtained by the LP.

Decision variables for these two ROO problems are reservoir releases in each operating period (decision point). The objective is to maximize the benefit from the system over 12 2-h operating periods, as:

$$Maximize \quad F = \sum_{i=1}^{N}\sum_{t=1}^{T} a_i(t) \times Re_i(t) \tag{15}$$

in which, i is the counter of the reservoir number; N is the total number of reservoirs; t is the counter of operation period; T is the total of operation period; $a_i(t)$ is the benefit per unit of release of reservoir i in period t; $Re_i(t)$ is the release of reservoir i in period t.

The operation of each reservoir is based on the continuity constraint. This equation over operating period t for reservoir i is as the following equation:

$$S_i(t+1) = S_i(t) + I_i(t) + M\, Re_i(t) \quad \forall t = 1,...,T \quad \& \quad i = 1,...,N \tag{16}$$

in which, $S_i(t)$ is the storage of reservoir i at the start of period t; $I_i(t)$ is the inflow to reservoir i in period t; M is a $N \times N$ matrix of lists of reservoir connectivity (the relations between reservoirs based on inflow and outflow).

The matrixes reservoir connectivity for the 4- and 10-reservoir problems are:

$$M = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix} , \quad for\ four-reservoir \tag{17}$$

$$M = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix} , \quad for\ ten-reservoir \tag{18}$$

The numbers 1 and -1 in the matrices show inflow and outflow to the reservoirs, respectively. Constraints related to reservoir releases and storages are as follows:

$$Re_i^{\min} \le Re_i(t) \le Re_i^{\max} \quad \forall t = 1,...,T \quad \& \quad i = 1,...,N \tag{19}$$

$$S_i^{\min} \le S_i(t) \le S_i^{\max} \quad \forall t = 1,...,T \quad \& \quad i = 1,...,N \tag{20}$$

$$S_i(T+1) = S_i(1) \quad \forall i = 1,...,N \tag{21}$$

where $Re_i^{\min}$ is the minimum release of reservoir i in period t; $Re_i^{\max}$ is the maximum release of reservoir i in period t; $S_i^{\min}$ is the minimum storage of reservoir i in period t; $S_i^{\max}$ is the maximum storage of reservoir i in period t.

The constant parameters of Eqs. (19)-(21) for the 4- and 10-reservoir problems are:

$$Re_i^{\min} = 0.005 \,, Re_i^{\max} = \begin{bmatrix} 4 \\ 4.5 \\ 4.5 \\ 8 \end{bmatrix}, S_i^{\min} = 1, S_i(1) = \begin{bmatrix} 6 \\ 6 \\ 6 \\ 8 \end{bmatrix} \,, for\ four - reservoir \tag{22}$$

$$Re_i^{\min} = \begin{bmatrix} 0.005 \\ 0.005 \\ 0.005 \\ 0.005 \\ 0.006 \\ 0.006 \\ 0.01 \\ 0.008 \\ 0.008 \\ 0.01 \end{bmatrix}, Re_i^{\max} = \begin{bmatrix} 4 \\ 4.5 \\ 2.12 \\ 7 \\ 6.43 \\ 4.21 \\ 17.1 \\ 3.1 \\ 4.2 \\ 18.9 \end{bmatrix}, S_i^{\min} = \begin{bmatrix} 1 \\ 1 \\ 0.3 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0.5 \\ 1 \end{bmatrix}, S_i(1) = \begin{bmatrix} 6 \\ 6 \\ 3 \\ 8 \\ 8 \\ 7 \\ 15 \\ 6 \\ 5 \\ 15 \end{bmatrix} \,, for\ ten - reservoir \tag{23}$$

Other values such as $S_i^{\max}$, $a_i(t)$ and $I_i(t)$ are as presented in Ref. [29].

If the reservoir storage does not meet the constraints in Eqs. (20) and (21), the results are infeasible and a penalty function should be applied. For the ultimate target function, the penalty functions are used as follows:

$$P1_i(t) = K_1 \left(S_i^{\min} - S_i(t) \right)^2 \quad t = 1,...,T \,\&i = 1,...,N \tag{24}$$

$$P2_i(t) = K_2 \left(S_i(t) - S_i^{\max} \right)^2 \quad t = 1,...,T \,\&i = 1,...,N \tag{25}$$

$$P3_i = K_3 \left(S_i(T+1) - S_i(1) \right)^2 \quad i = 1,...,N \tag{26}$$

where P are penalty functions, K_1, K_2, K_3 are penalty constants were considered equal to 60, 40, and 40, respectively.

At the end, the objective function is based on the following equation:

$$Maximise \quad F = \sum_{i=1}^{N}\sum_{t=1}^{T}\left[a_i(t)\times Re_i(t)-\left(P1_i(t)+P2_i(t)+P3_i(t)\right)\right] \qquad (27)$$

6. RESULTS AND DISCUSSIONS

In this study, in order to evaluate the performance of the CBO, ECBO and VPS algorithms [30, 31], the well-known hypothetical multi-reservoir optimization problems are used. All the algorithms are programmed with *Matlab 8.5* on a Windows 7 professional using an Intel Core i7-M 620, 2.67 GHz and 4 GB RAM computer.

The global optimal objective functions of the problems are obtained with non-linear programming solution by the *LINGO 17.0* software and are equal to 308.29 and 1194.44 for four- and ten-reservoir systems, respectively. These values are the same with those reported in Ref. [14].

The CBO is a non-parameter algorithm but the values of the ECBO and VPS parameters were obtained in a sensitivity analysis process. Wherein the results of different runs with different combination of parameters are compared and the best parameters values are chosen. For fairly comparison between proposed algorithms, the random initial solutions of each runs, the maximum number of iterations and the population size are the same. The parameters of the algorithms used in this study are listed in Table 1. The results obtained for the optimization of two ROO problems are presented in the next two following sections.

Table 1: Parameters of used algorithms

Algorithm	Parameter		Value
CBO	-		-
ECBO	Colliding memory (*CM*)		*n*/4
	Regeneration probability (*Pro*)		0.3
	α		0.05
	w_1		0.3
VPS	w_2		0.3
	p		0.7
	HMCR		0.95
	PAR		0.1
Common parameters	Number of populations (*n*)	Four-Reservoir	200
		Ten-Reservoir	250
	Maximum number of iterations (*Iter$_{max}$*)	Four-Reservoir	2500
		Ten-Reservoir	5000

6.1 Results of the four-reservoir problem

The optimal solution reported by Chow and Cortez-Rivera [28] is equal to 308.26. The best solution achieved by Murray and Yakowitz [29] used differential dynamic programming (DDP) is equal to 308.23 and 307.98 after eight and 20 iterations, respectively.

Recently, several researchers solved four-reservoir operation optimization model by employing different types of meta-heuristic algorithms. A literature review on this problem is given in Table 2.

Table 2: Different results obtained for the four-reservoir problem

Study	Algorithm	Ave[a]	SD[b]	Npop[c]	MNFE[d]	MaxIt[e]
	HBMO [14]	307.50	0.417	220	1,100,000	-
	BA [16]	307.84	0.350	50	500,050	-
Bozorg-Hadad et al.	WCA [17]	304.92	1.887	100	-	5,000
	BBO [19]	307.69	0.511	-	500,000	-
	GSA [21]	308.30	0.277	200	500,000	-
	GA [21]	299.60	0.705	200	500,000	-
	GA	301.54	-	50	-	20,000
Hosseini-Moghari et al. [18]	ICA	305.11	-	50	-	20,000
	COA	306.90	-	50	-	20,000
Asgari et al. [20]	GA	299.69	0.689	200	-	8,000
	WOA	307.75	0.364	-	-	20,000
Ahmadianfar et al. [32]	IBA	308.05	0.150	300	-	780
Garousi-Nejad et al. [33]	FA	305.51	0.665	50	500,050	-
	MFA	308.21	0.050	50	500,050	-
Solgi et al. [34]	EHBMO	308.08	0.321	211	-	4,000
	GA	306.72	0.580	-	50,000	-
Ehteram et al. [35]	Krill	307.26	0.220	-	50,000	-
	Hybrid	308.17	0.050	-	50,000	-

[a] The average value of the objective functions obtained by algorithms.
[b] Standard division of the objective function values.
[c] The number of population.
[d] The maximum number of function evaluations executed to achieve the optimal solution.
[e] The maximum number of iteration.

Table 2 indicates that GSA algorithm [41] achieved the nearest average solution to the absolute global optimum (NLP) compared with all other EAs. It is concluded that the hybrid GA and krill algorithms that was introduced by Ehteram et al. [44] solved the problem with the lowest maximum number of functional evaluations (50,000) compared with other algorithms and obtained 308.17 as the average objective function value that close to 99.96% of the global optimum.

For the purpose of testing the performance of the CBO, ECBO and VPS algorithms in this study, this problem is solved with 200 and 2500 as the number of population size and the maximum number of iteration, respectively. The results of 10 independent runs of all presented algorithms are shown in Table 3 and Fig. 2.

Table 3: Results of 10 runs for the four-reservoir problem

No. of run	CBO	ECBO	VPS
1	277.70	305.86	305.39
2	274.88	305.22	306.54
3	279.66	306.90	307.42
4	279.49	303.91	305.82
5	275.92	306.35	306.82
6	274.47	303.91	305.26
7	277.77	305.47	306.39
8	276.35	305.19	305.66
9	273.68	306.66	305.82
10	280.27	305.50	306.93
Best	280.27	306.90	307.42
Worst	273.68	303.91	305.26
Average	277.02	305.50	306.21
Standard Deviation	2.32	1.02	0.72
Coefficient of Variation	0.0084	0.0033	0.0024
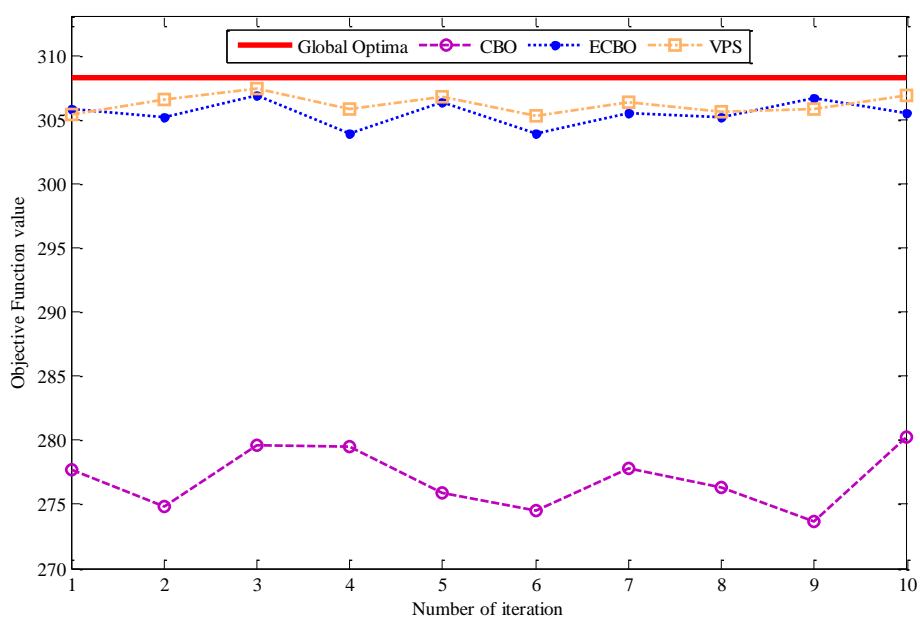
Figure 2. The results of 10 different runs of CBO, ECBO and VPS for the four-reservoir operation

As is shown in Table 3, the VPS algorithm achieved a closer solution to the global optimum (NLP solution) in contrast with CBO and ECBO. According to Table 3, the best result out of ten different runs was achieved by the VPS is 307.42 which is close to global optimum up to 99.72%. The average value of objective function obtained by the VPS exhibits 0.675% difference with the global optimal solution of the problem. In addition, the worst solution obtained by the VPS is better than the worst solution reaches by other

methods. The small value of the standard deviation and coefficient of variation for the VPS illustrates the superior capacity of this algorithm to reach a close point to the global optimum. After that, ECBO shows better performance in contrast with CBO by reaching 306.90, 305.50 and 303.91 as the best, average and worst solution, respectively.

Figs. 3-5 show the convergence rates versus the number of function evaluations of the best, worst and average solutions of the three algorithms in 10 runs. The results show that VPS has better and faster convergence than other algorithms towards the optimal solution where it is established the superior performance of the VPS relative to these methods.
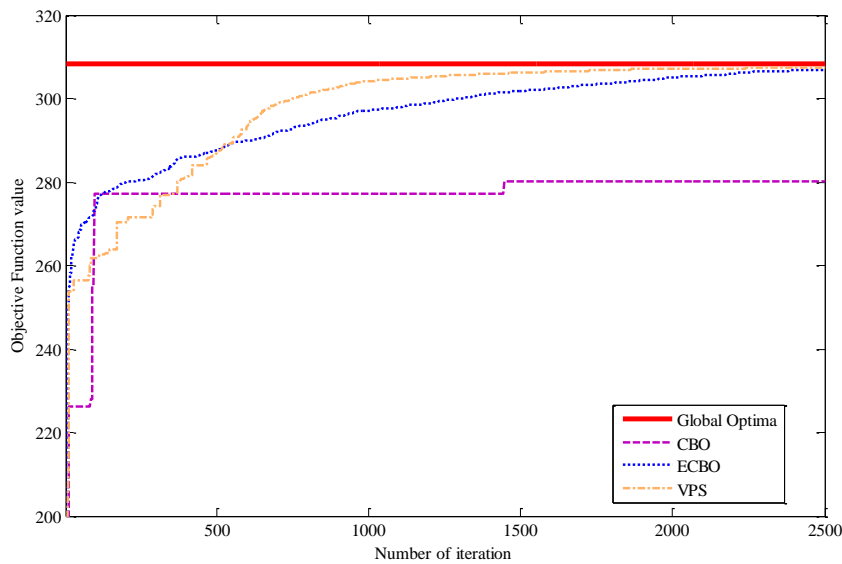
Figure 3. The best objective value for the four-reservoir operation
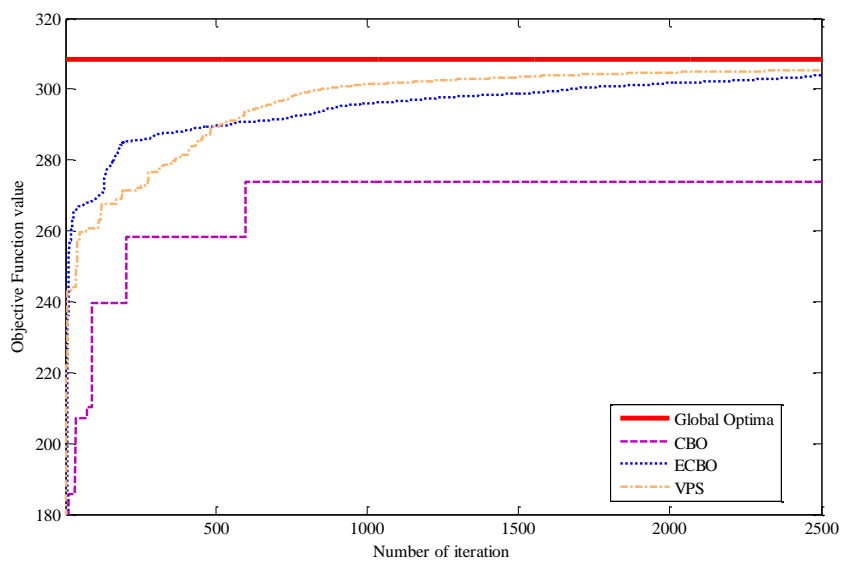
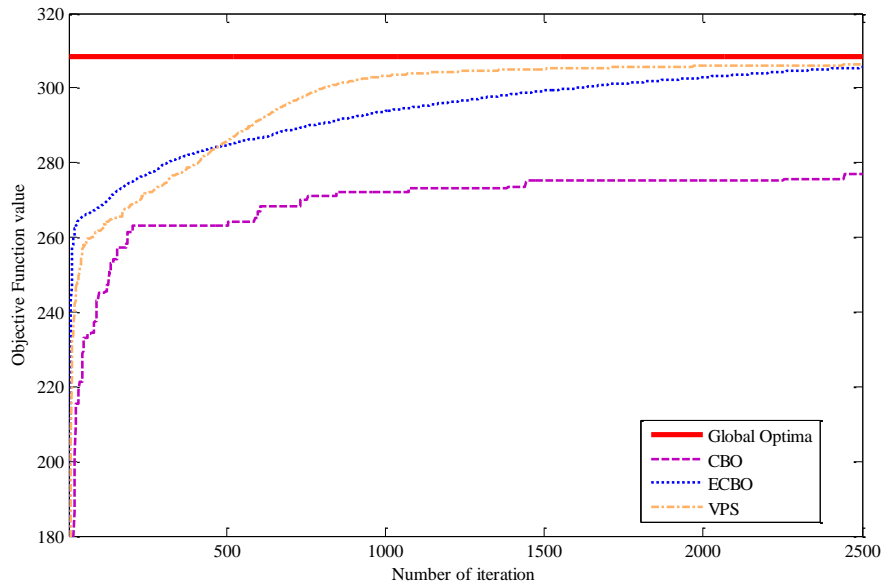Figure 4. The worst objective value for the four-reservoir operation

Figure 5. The average objective value for the four-reservoir operation

Figs. 6 and 7 show the monthly reservoir release and reservoir storage patterns of the best solution for the operation of the four-reservoir using the LP with VPS and ECBO as two superior algorithms. It can be seen that there is no any constraint violation and also the linear programming is almost entirely compatible with the VPS algorithm.
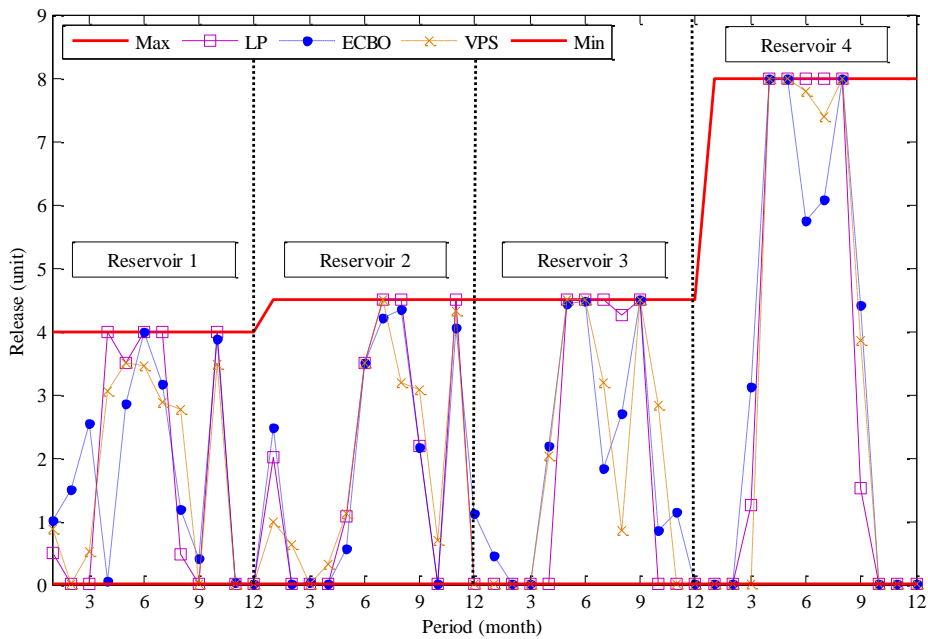

Figure 6. Comparison between the best result of ECBO, VPS and LP for reservoir releases
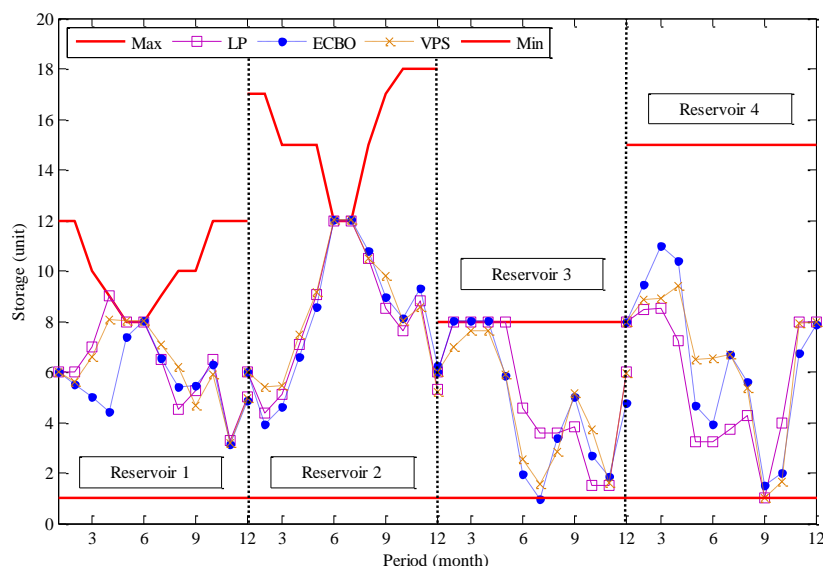
Figure 7. Comparison between the best result of ECBO, VPS and LP for reservoir storages

6.2 Results of the ten-reservoir problem

Murray and Yakowitz [29] employed differential dynamic programming (DDP) and they reported 1190.625 as the best solution for this problem that get as close as 99.8% of the global optimum obtained from LP using Lingo software (1194.44). Table 4 shows the literature review on the results of this problem obtained by different meta-heuristics.

From Table 4, the multi-colony ACO algorithm find an optimum result with the lowest number of functional evaluations (450,000) compared to the other algorithms. The optimal objective function values were 1190.25, 1148.05, 1193.91, 1192.89, 1183.59 and 1185. 22 for GA, HBMO, hybrid GA and krill, IBA, MFA and multi-colony ACO algorithm, respectively. The highest percentage of the objective function relative to the absolute global optimal solution is 99.95%, obtained by Ehteram et al. [44].

Table 4: Obtained results for the 10-reservoir system

Study	Algorithm	Ave[a]	SD[b]	Npop[c]	MNFE[d]	MaxIt[e]
Wardlaw and Sharif [6]	GA	1190.25	-	500	-	2500
Jalali et al. [12]	ACO	1185.22	3.60	150	-	3,000
Bozorg-Hadad et al. [14]	HBMO	1148.05	5.00	220	1,320,000	-
Ahmadianfar et al. [32]	IBA	1192.89	0.69	100	-	9700
Garousi-Nejad et al. [33]	FA	1097.41	8.0142	50	1,000,000	-
	MFA	1183.59	1.5177	50	1,000,000	-
Ehteram et al. [35]	GA	1188.68	1.13	-	500,000	-
	Krill	1189.66	0.707	-	500,000	-
	Hybrid	1193.91	0.31	-	500,000	-

[a] The average value of the objective functions obtained by algorithms.

[b] Standard division of the objective function values.

[c] The number of population.

[d] The maximum number of function evaluations executed to achieve the optimal solution.

[e] The maximum number of iteration.

In this study, 10 different runs of the three algorithms are carried out with 250 and 5,000 as population size and the maximum number of iterations, respectively. The related results are shown in Table 5 and Fig. 8.

Table 5: Results for 10 runs for the ten-reservoir problem

| No. of run	CBO	ECBO	VPS
1	1017.13	1176.04	1184.26
2	1062.00	1180.60	1183.48
3	1051.65	1179.46	1180.47
4	1070.38	1179.27	1184.50
5	1071.48	1181.54	1180.47
6	1095.31	1174.23	1184.50
7	1029.58	1175.24	1181.62
8	1042.51	1183.37	1177.26
9	1089.82	1182.25	1176.38
10	1081.46	1187.40	1189.19
Best	1095.31	1187.40	1189.19
Worst	1017.13	1174.23	1176.38
Average	1061.13	1179.94	1182.21
Standard Deviation	25.75	4.03	3.80
Coefficient of Variation	0.0243	0.0034	0.0032
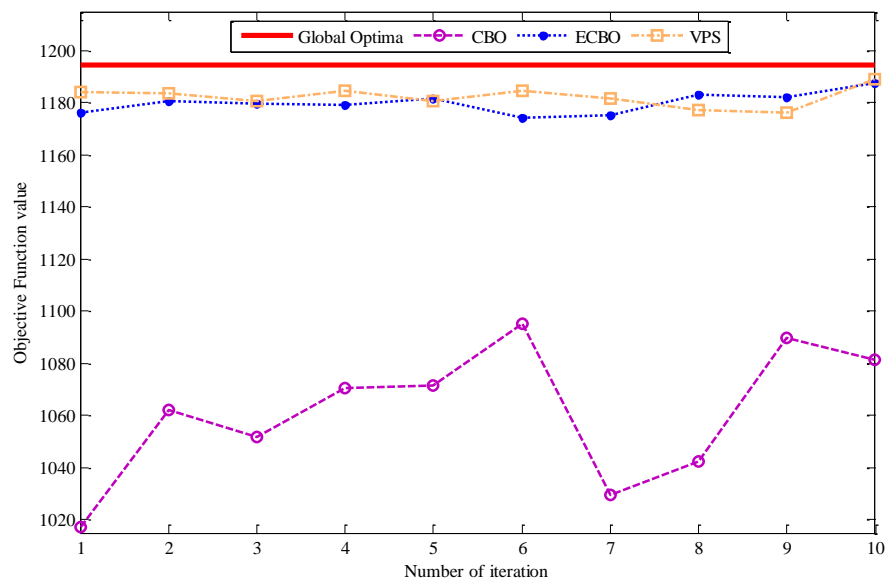
Figure 8. The results of 10 runs of CBO, ECBO and VPS for the ten-reservoir operation

The relative error associated with the average value of the objective function for the CBO, ECBO and VPS are about 11.16%, 1.21% and 1.02% compared to global solution (the LP result), respectively. The results of Table 5 show that, in addition to a suitable performance of the VPS in reaching the global optimal solution, the standard deviation of the objective function value and coefficient of variation of 10 different runs are equal to

3.80 and 0.0032, respectively. The standard deviation of the objective function value obtained by the CBO is equal to 25.75, which is approximately 6.8 times larger than that of VPS in 10 runs. The best, average and worst values of the objective function of VPS are 99.56%, 98.98% and 98.49% of the global optimal solution (1194.44), respectively.

Convergence rate of the best, worst, and average solution over the 10 runs are presented in Figs. 9, 10 and 11. They show the superior performance of the VPS compared to other methods. As it can be seen, the VPS shows better and even faster convergence towards the optimal solution.

To be more informative, monthly releases and storages from different reservoirs, along with allowable range of releases and storages are presented in Figs. 12 and 13.
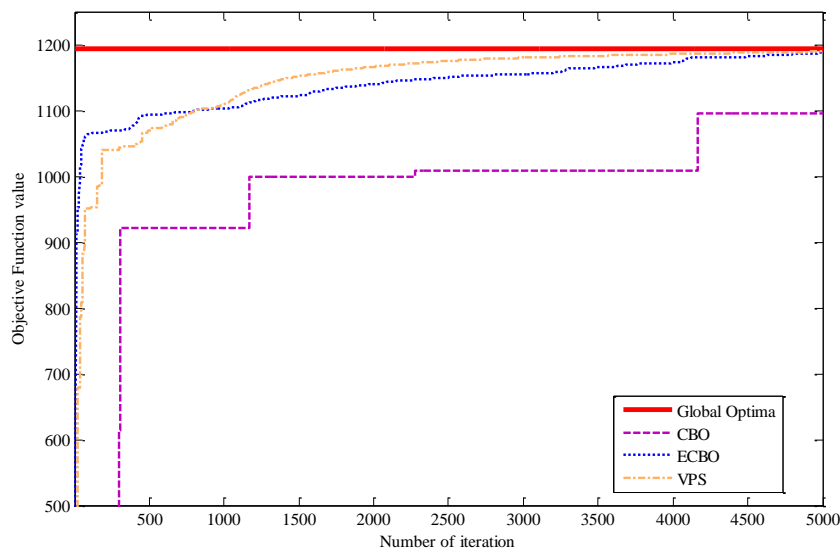

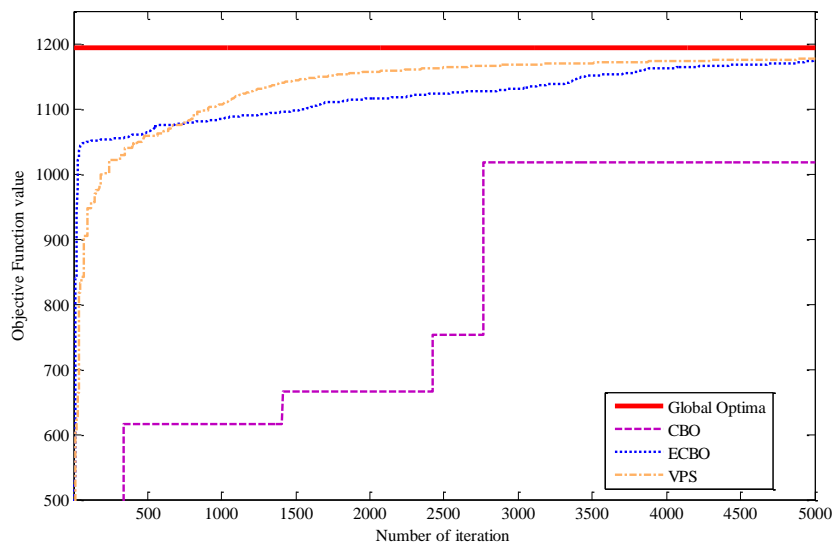Figure 9. The best objective value for the ten-reservoir operation

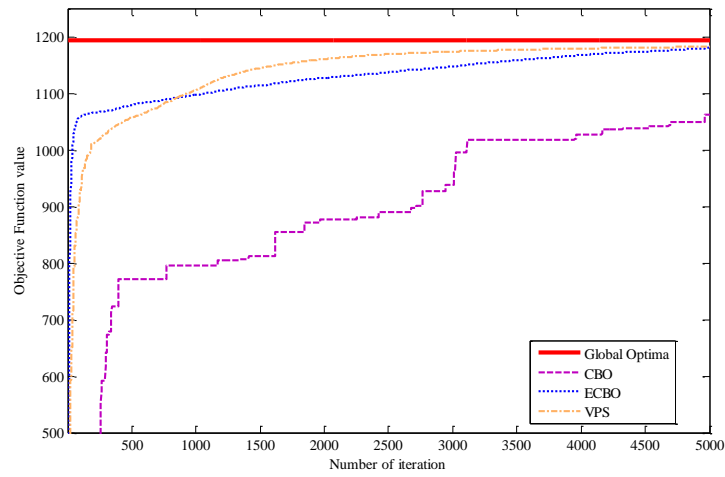Figure 10. The worst objective value for the ten-reservoir operation

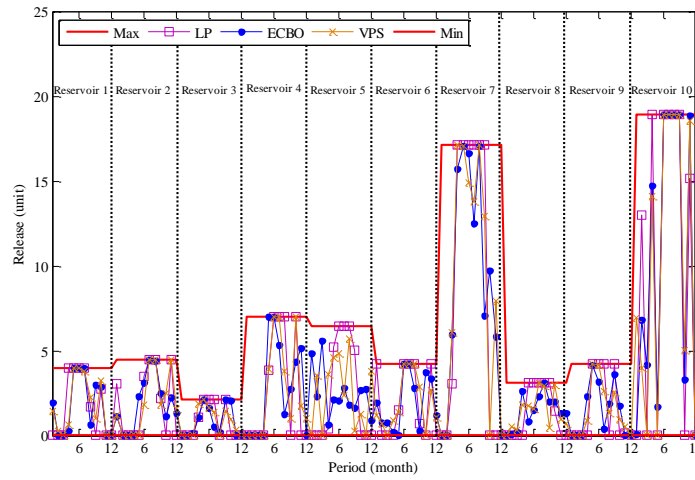Figure 11. The average objective value for the ten-reservoir operation

Figure 12. Comparison between the best result of ECBO, VPS and LP for reservoir releases
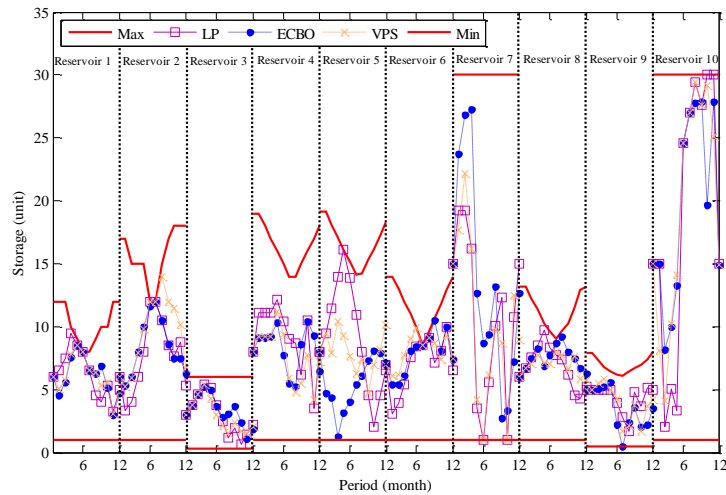

Figure 13. Comparison between the best result of ECBO, VPS and LP for reservoir storages

7. CONCLUDING REMARKS

In this study, the capability of the three new developed meta-heuristic algorithms was evaluated in solving two well-known reservoir operation optimization benchmark problems (i.e. four- and ten-reservoir systems). Results indicate the high ability of the VPS to solve these problems. According to the results of different runs, the best solutions obtained by the VPS for four-reservoir and ten-reservoir system are 307.42 and 1189.19 which are close to global optimum obtained from LP up to 99.72% and 99.56 %, respectively. Comparing the results of the VPS with two other algorithms (CBO and ECBO), it can be concluded that the VPS can solve the considered problems with less computational efforts and a fast convergence rate. So, it proves the high capability of the VPS to solve large scale reservoirs system operation problems. After that ECBO has been more successful in approaching the global optimum solution obtained from *LINGO 17.0*. Although, parameter independency is an important advantage of the CBO algorithm but results indicate the superiority of its enhanced version.

In addition, the lowest standard deviation and coefficient of variation obtained by the VPS show a better performance, higher accuracy and faster convergence to a given solution in each run of this algorithm. So, the VPS has superiority and it can be stated that the VPS is a reliable tool for reservoir operation optimization problems.

REFERENCES

1. Stedinger JR, Sule BF, Loucks DP. Stochastic dynamic programming models for reservoir operation optimization, *Water Res Res* 1984; **20**(11): 1499-1505.
2. Hall WA, Butcher WS, Esogbue A. Optimization of the operation of a multiple purpose reservoir by dynamic programming, *Water Res Res* 1968; **4**(3): 471-7.
3. Yakowitz S. Dynamic programming applications in water resources, *Water Res Res* 1982; **18**(4): 673-96.
4. Kelman J, Stedinger JR, Cooper LA, Hsu E, Yuan SQ. Sampling stochastic dynamic programming applied to reservoir operation, *Water Res Res* 1990; **26**(3): 447-54.
5. Hossain MS, El-Shafie A. Intelligent systems in optimizing reservoir operation policy: a review, *Water Res Manag* 2013; **27**(9): 3387-3407.
6. Wardlaw R, Sharif M. Evaluation of genetic algorithms for optimal reservoir system operation, *J Water Res Plann Manag* 1999; **125**(1): 25-33.
7. Cai X, McKinney DC, Lasdon LS. Solving nonlinear water management models using a combined genetic algorithm and linear programming approach, *Adv Water Res* 2001; **24**(6): 667-76.
8. Chen L. Real coded genetic algorithm optimization of long term reservoir operation, *J American Water Res Associat* 2003; **39**(5): 1157-65.
9. Ahmed JA, Sarma AK. Genetic algorithm for optimal operating policy of a multipurpose reservoir, *Water Res Manag* 2005; **19**(2): 145-61.
10. Kumar DN, Reddy MJ. Ant colony optimization for multi-purpose reservoir operation, *Water Res Manag* 2006; **20**(6): 879-98.

11. Jothiprakash V, Shanthi G. Single reservoir operating policies using genetic algorithm, *Water Res Manag* 2006; **20**(6): 917-29.
12. Jalali MR, Afshar A, Marino MA. Multi-colony ant algorithm for continuous multi-reservoir operation optimization problem, *Water Res Manag* 2007; **21**(9): 1429-47.
13. Zahraie B, Hosseini SM. Development of reservoir operation policies considering variable agricultural water demands, *Expert Syst Appl* 2009; **36**(3): 4980-7.
14. Bozorg-Haddad O, Afshar A, Mariño MA. Multireservoir optimisation in discrete and continuous domains, *Proc Inst Civil Eng Water Manag* 2011; **164**(2): 57-72.
15. Dariane AB, Sarani S. Application of intelligent water drops algorithm in reservoir operation, *Water Res Manag* 2013; **27**(14): 4827-43.
16. Bozorg-Haddad O, Karimirad I, Seifollahi-Aghmiuni S, Loáiciga HA. Development and application of the bat algorithm for optimizing the operation of reservoir systems, *J Water Res Plann Manag* 2014; **141**(8): 04014097.
17. Bozorg-Haddad O, Moravej M, Loáiciga HA. Application of the water cycle algorithm to the optimal operation of reservoir systems, *J Irrigat Drain Eng* 2014; **141**(5): 04014064.
18. Hosseini-Moghari SM, Morovati R, Moghadas M, Araghinejad S. Optimum operation of reservoir using two evolutionary algorithms: imperialist competitive algorithm (ICA) and cuckoo optimization algorithm (COA), *Water Res Manag* 2015; **29**(10): 3749-69.
19. Bozorg-Haddad O, Hosseini-Moghari SM, Loáiciga, HA. Biogeography-based optimization algorithm for optimal operation of reservoir systems, *J Water Res Plann Manag* 2015; **142**(1): 04015034.
20. Asgari HR, Bozorg Haddad O, Pazoki M, Loáiciga HA. Weed optimization algorithm for optimal reservoir operation, *J Irrigat Drain Eng* 2015; **142**(2): 04015055.
21. Bozorg-Haddad O, Janbaz M, Loáiciga HA. Application of the gravity search algorithm to multi-reservoir operation optimization, *Adv Water Res* 2016; **98:** 173-85.
22. Moravej M, Hosseini-Moghari SM. Large Scale Reservoirs System Operation Optimization: the Interior Search Algorithm (ISA) Approach, *Water Res Manag* 2016; **30**(10): 3389-3407.
23. Kaveh A, Mahdavi VR. Colliding bodies optimization: A novel meta-heuristic method, *Comput Struct* 2014; **139:** 18-27.
24. Kaveh A, Ilchi Ghazaan M. Enhanced colliding bodies optimization for design problems with continuous and discrete variables, *Adv Eng Softw* 2014; **77:** 66-75.
25. Kaveh A, Ilchi Ghazaan M. A new meta-heuristic algorithm: vibrating particles system, *Sci Iran, Transactions A, Civil Eng* 2017; **24**(2): 551-66.
26. Kaveh A, Ilchi Ghazaan M. A comparative study of CBO and ECBO for optimal design of skeletal structures, *Comput Struct* 2015; **153**: 137-47.
27. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, 2nd edition, Springer, Switzerland, 2017.
28. Chow VT, Cortes-Rivera G. Application of DDDP in water resources planning, Research Rep, University of Illinois, Urbana-Champaign, IL, 1974.
29. Murray DM, Yakowitz SJ. Constrained differential dynamic programming and its application to multireservoir control, *Water Res Res* 1979; **15**(5): 1017-27.
30. Kaveh A, Ilchi Ghazaan M. Computer codes for colliding bodies optimization and its enhanced version, *Int J Optim Civil Eng* 2014; **4**(3): 321-32.

31. Kaveh A, Ilchi Ghazaan M. Matlab code for Vibrating Particles System algorithm, *Int J Optim Civil Eng* 2017; **7**(3): 355-66.
32. Ahmadianfar I, Adib A, Salarijazi M. Optimizing multireservoir operation: Hybrid of bat algorithm and differential evolution, *J Water Res Plann Manag* 2015; **142**(2): 05015010.
33. Garousi-Nejad I, Bozorg-Haddad O, Loáiciga HA. Modified firefly algorithm for solving multireservoir operation in continuous and discrete domains, *J Water Res Plann Manag* 2016; **142**(9): 04016029.
34. Solgi M, Bozorg-Haddad O, Loáiciga HA. The Enhanced Honey-Bee Mating Optimization Algorithm for Water Resources Optimization, *Water Res Manag* 2016; **3**(31): 885-901.
35. Ehteram M, Mousavi SF, Karami H, Farzin S, Emami M, Othman FB, Amini Z, Kisi O, El-Shafie A. Fast convergence optimization model for single and multi-purposes reservoirs using hybrid algorithm, *Adv Eng Inform* 2017; **32:** 287-98.