A hybrid constrained coral reefs optimization algorithm with machine learning for optimizing multi-reservoir systems operation

Mohammad Emami [ac], Sara Nazif [b], Sayed-Farhad Mousavi [a], Hojat Karami [a], Andre Daccache [c]

[a] Department of Water Engineering and Hydraulic Structures, Faculty of Civil Engineering, Semnan University, Semnan, Iran; emami.m@semnan.ac.ir

[b] School of Civil Engineering, College of Engineering, University of Tehran, Tehran, Iran;

[c] Department of Biological and Agricultural Engineering, University of California, Davis (UC Davis), CA, USA;

Abstract

The continuous growing demand for water, prolonged periods of drought, and climatic uncertainties attributed mainly to climate change mean surface water reservoirs more than ever need to be managed efficiently. Several optimization algorithms have been developed to optimize multi-reservoir systems operation, mostly during severe dry/wet seasons, to mitigate extreme-events consequences. Yet, convergence speed, presence of local optimums, and calculation-cost efficiency are challenging while looking for the global optimum. In this paper, the problem of finding an efficient optimal operation policy in multi-reservoir systems is discussed. The complexity of the long-term operating rules and the reservoirs' upstream and downstream joint-demands projected in recursive constraints make this problem formidable. The original Coral Reefs Optimization (CRO) algorithm, which is a meta-heuristic evolutionary algorithm, and two modified versions have been used to solve this problem. Proposed modifications reduce the calculation cost by narrowing the search space called a constrained-CCRO and adjusting reproduction operators with a reinforcement learning approach, namely the Q-Learning method (i.e., the CCRO-QL algorithm). The modified versions search for the optimum solution in the feasible region instead of the entire problem domain. The models' performance has been evaluated by solving five mathematical benchmark problems and a well-known continuous four-reservoir system (CFr) problem. Obtained results have been compared with those in the literature and the global optimum, which Linear Programming (LP) achieves. The CCRO-QL is shown to be very calculation-cost-effective in locating the global optimum or near-optimal solutions and efficient in terms of convergence, accuracy, and robustness.

Keywords

Water resources management, Artificial intelligence, Heuristic method, Decision support tool, Multi-agent approach, Particle swarm optimization

1. Introduction

Reservoirs are usually designed to serve multiple users: urban, environmental, industrial, agricultural, and hydropower. Optimizing a reservoir's operation while considering the quantitative flow characteristics and water demands along with climatic conditions is a complex process that deals with several parameters. Adding these parameters to long-term planning and management would result in a variety of decision criteria and objective functions that include hundreds of variables and constraints.

In a multi-reservoir water supply system, joint operating rules should address the total release from the system and the amounts to be released from each reservoir in all periods according to the reservoir's upstream and downstream demands. This optimization problem in water resources systems is too complex to be solved using classical optimization methods. With dimensionality and non-linearity as the primary causes of complexity, researchers have now used heuristic optimization methods to derive policies that best serve multiple water uses and help identify the tradeoffs between various objectives. Since classical optimization approaches are limited in managing search space structure, modern meta-heuristic methods have been the core of recent research dealing with "hard" optimization problems. For example, bioinspired algorithms have been developed that use the aspects of natural evolution, based on the continued survival of the fittest individuals, to find an optimal or near-optimal solution.

Ehteram et al. (2017a, b) applied the Shark Algorithm and a new hybrid algorithm by combining the GA with the Krill Algorithm to maximize the generated hydropower and water supply benefits in four-reservoir and ten-reservoir benchmark systems [16,17]. Samadi-koucheksaraee et al. (2019) applied a Gradient Evolution (GE) algorithm to optimize single- and multi-reservoir systems. GE is employed to optimize several case studies demonstrating the proposed method's superior ability [41]. Mohammadi et al. (2019) applied the TOPSIS technique to compare the genetic algorithm's performance with a Hybrid Whale-Genetic Algorithm (HWGA) for optimizing multi-reservoir systems, which shows the advantage of the hybrid method over GA and Whale Optimization Algorithm (WOA) alone [27]. In the most recent studies [7,26], a constrained version of the Improved Artificial Bee Colony (IABC) algorithm and a hybrid Cellular Automata-Simulated Annealing (CA-SA) method implemented to optimize reservoir operation and hydropower operation, respectively. Both proposed methods show relative improvements in benchmark problems and real-world case studies. Yet evolutionary optimization algorithms significantly contribute to flood control and susceptibility assessment [13,29] and uncertainty assessment methods [4,5].

A wide variety of meta-heuristic methods have been proposed and investigated since they are approximate and usually non-deterministic. However, meta-heuristic algorithms do not guarantee a globally optimal solution, especially in certain classes of problems. As such, further approaches are being investigated, which may provide a sufficiently good solution to an optimization problem, especially for complex problems for which limited computational capacity is available [44].

As a novel bio-inspired algorithm, the *Coral Reefs Optimization* (CRO) algorithm was first introduced in 2013 by Salcedo-Sanz et al. [34,37]. This new method has been used to solve several continuous and discrete benchmark problems and practical ones.

Yang et al. (2016) reduced the prediction time and obtained better performance for training original ELM by hybridizing Differential Evolution (DE) with CRO [50]. Salcedo-Sanz et al. (2017) proposed the CRO with the substrate layer (CRO-SL) as a competitive co-evolution algorithm to control structures' vibration via tuned mass dampers. This approach provides benefits by exploiting the combination of different types of searching mechanisms [32]. Some recent research has improved the CRO's efficiency using hybridizing techniques to modify exploration and exploitation abilities. Most of them have focused on CRO-SL as a co-evolutionary algorithm [9,23,31,43]. Some others have combined CRO with: game theory [18], statistical approaches [14], simulated annealing [48], and memetic strategy [15], while some are still using the original version of CRO [3,6,19].

An expanding list of the CRO algorithm applications on top of generating reliable results in various fields reveals its potential capability in water resources optimization problems. At the same time, the proposed modifications are another confirmation of the correctness of the *No Free Lunch* (NFL) theorems in search [47] and optimization [46] problems. While almost all previous studies have focused on problem-solving methods, current research has also attempted to address how to describe the problem mathematically.

This study is the first to apply the CRO algorithm to solve water resources system optimization-problems to the best of the authors' knowledge. This work investigates the CRO algorithm's effectiveness in reservoir operation problems by solving a hypothetical multi-reservoir system. Two new modifications to the algorithm are proposed to reduce the calculation cost by narrowing the search space when the algorithm deals with recursive equations. Moreover, the reproduction operators have adjusted by applying a reinforcement learning approach (i.e., the Q-Learning method). This method can improve solutions in terms of convergence, accuracy, and calculation costs. The proposed model's performance was investigated by solving five mathematical benchmark problems and applying them to a multi-reservoir system.

2. Methodology

2.1. Coral Reefs Optimization (CRO) model description

The CRO is a novel meta-heuristic search approach based on coral reef formation and reproduction [40]. It simulates different types of corals reproduction (i.e., internal/external and sexual/asexual) and reef formation, including competition between new corals to settle in the reef and depredation of the weak ones.

The algorithm is based on a coral reef's artificial modeling, (Λ) consisting of an M×N square grid. Each square (i,j) can allocate a coral ($\Xi_{i,j}$), representing a possible solution to the optimization problem at hand. The primary reef forms by assigning some squares in Λ to be occupied by corals that are random solutions to the problem, and some other squares in the grid to be empty where new corals (polyps) in the next generations can freely settle and grow [34,35].

Each coral is associated with a function that denotes its *health* status, indicating the fitness of the solution at hand ($f(\Xi_{i,j})$: $\mathscr{I}\rightarrow\mathbb{R}$). The reef will progress if better corals or fitter solutions outlive the inferior ones. In iterative-based generations, new corals reproduce by sequentially applying several operators until a given stop criterion is met, for instance, a maximum number of iterations or a maximum number of objective function evaluations [40]. Such operators are used for modeling sexual reproduction (broadcast spawning and brooding), asexual reproduction (budding), and polyps depredation.

2.1.1. Broadcast spawning (External Sexual Reproduction, ESR)

The ESR operation consists of the following steps [30,36]:

1. At a given step *k* of the reef formation phase, a fraction of the existing corals is selected uniformly at random to be broadcast spawners (F_b). Corals that are not selected for broadcast spawning will reproduce later via brooding during the algorithm execution.
2. Each of the couples permitted to be parents will breed one or two coral larvae (depending on the operator) by sexual crossover. The couple selection can be either uniform random or any fitness proportionate selection approach.

2.1.2. Brooding (Internal Sexual Reproduction, ISR)

The fraction of corals that will reproduce via brooding is $1\text{-}F_b$. The modeling consists of forming a coral larva by a random mutation in the brooder coral (i.e., self-fertilization in hermaphrodite corals). The produced larvae are then released into the water as well as the offspring generated in step 2 [33].

2.1.3. Larvae setting

Once all larvae are formed at step *k* either through broadcast spawning or by brooding, they will try to settle and grow in the reef, based on their *health* function. The struggle for survival starts when each larva tries to settle on a random square (*i,j*) of the reef. If this square is empty, the coral grows therein independently of the value of its *health* function. Otherwise, when the given position is already occupied, the new larva will set if it is healthier than the existing coral. Each larva has a limited chance to compete with the existing ones. After κ unsuccessful attempts, it will be discarded and eliminated from the population [34].

2.1.4. Budding or fragmentation

In this asexual reproduction process, a small fraction of healthier corals (F_a) duplicates itself and tries to settle in a different part of the reef, following the configuration process described in step 2.1.3. A predefined maximum number (μ) of equal corals are allowed in the reef. Mutations are also introduced with probability P_a to obtain variability [40].

2.1.5. Depredation

A small number of corals (applying a very small probability P_d, exclusively to a fraction F_d of the least fit corals) drops at the end of each iteration k for the sake of liberating space in the reef for the next generation.

Figure 1 illustrates the flowchart of the CRO algorithm, along with the operators described above. External sexual reproduction (i.e., broadcast spawning process) has the primary role of exploration, whereas internal sexual reproduction (i.e., brooding) is essential for local search. The budding (or asexual reproduction) ensures that the best solutions replicate and span over the reef; so, this process facilitates the exploitation characteristic of the CRO algorithm while maintaining elitism.

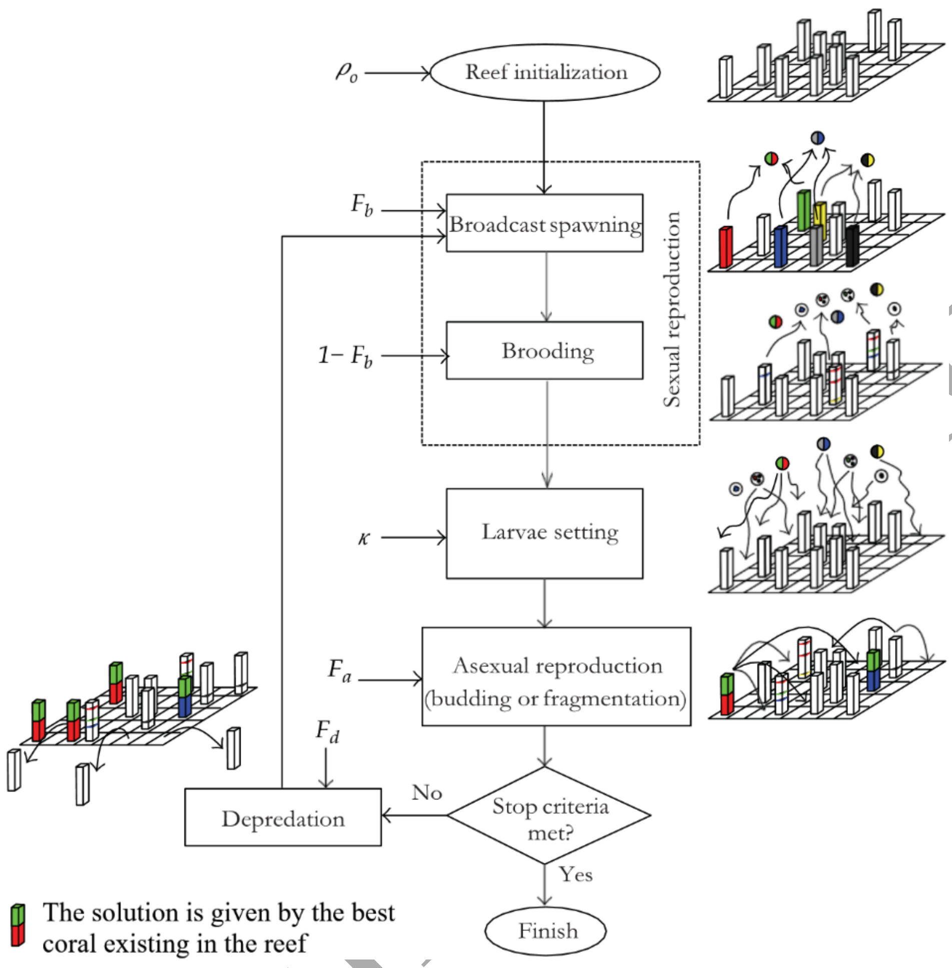


Figure 1: Flow diagram of the proposed CRO algorithm [40]

2.2. CRO evaluation in Continuous Benchmark Problems (CBP)

In applied mathematics, test functions, known as artificial landscapes, are used to evaluate optimization algorithms' characteristics. This includes the convergence rate, precision, robustness, and general performance. Five well-known continuous benchmark functions (Table 1) have been applied for evaluating the proposed CRO algorithm compared to other evolutionary algorithms [21]. Figure 2 illustrates the unimodality of all functions, while the f4 and f5 have many local minima. For the sake of visualization, a fixed number of variables (n=2) has been applied for f2 ~ f5. Table S.1 in Appendix A (Supplementary Information) illustrates all CRO parameters' values and the operators following the guidelines provided by Salcedo-Sanz et al. (2014). The parameters controlling the CRO model were identical to those in the previous study [40]. The number of function evaluations used was the same for all compared algorithms: 20,000 in f_1, f_2, f_4, and 10,000 in f_3 and f_5. The number of function evaluations (NFE), as a hardware-independent index, was used to evaluate and assess the computational cost (effort) and the complexity in different CRO versions.

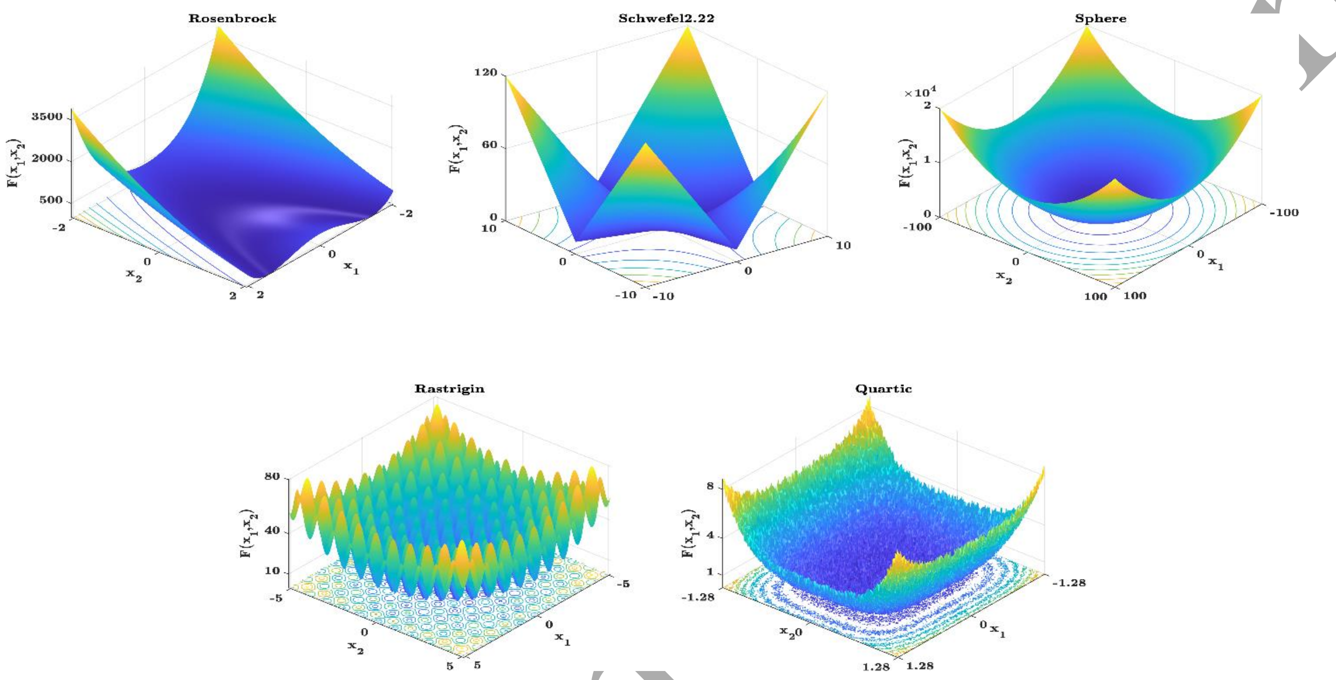


Figure 2: The five continuous mathematical functions in a 3-dimensional coordinate system (n=2)

Due to the non-deterministic nature of the studied algorithms, comparisons should be made on a large set of results obtained from each algorithm's independent executions. The Kruskal-Wallis test can validate the statistical significance of the obtained results by different algorithms. It is a non-parametric method as an extension to the Mann–Whitney U test for testing whether samples originate from the same distribution [45]. At each simulation, 30 executions of each algorithm are launched to obtain well-sampled performance statistics (best, average, and standard deviation for all iterations).

The original version of CRO takes advantage of Gaussian and Cauchy mutations (G+C) [40]. In this research, a simulated binary crossover (SBX) operator, which uses polynomial probability distribution ($\mathcal{P}$) [12] is studied for brooding of the corals.

Table 1. Summary of the CBP functions considered in the current study [51]

Test function	Expression	Number of variables (n)	Feasible region	Global optimum
f₁: Rosenbrock	$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	2	$[-2.048,2.048]^n$	$f_{min} = 0$ at (1,1)
f₂: Schwefel 2.22	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10,10]^n$	$f_{min} = 0$ at (0,0,…,0)
f₃: Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]^n$	$f_{min} = 0$ at (0,0,…,0)
f₄: Rastrigin	$f(x) = 10.n + \sum_{i=1}^{n} [x_i^2 - 10.\cos(2\pi x_i)]$	10	$[-5.12,5.12]^n$	$f_{min} = 0$ at (0,0,…,0)
f₅: Quartic	$f(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	30	$[-1.28,1.28]^n$	$f_{min} = 0 + rand$ at (0,0,…,0)

$$\mathcal{P}(\delta) = 0.5(\eta + 1)(1 - |\delta|)^{\eta} \qquad \delta \in (-1,1) \tag{1}$$

$$\delta = \begin{cases} (2u)^{\frac{1}{\eta+1}} - 1, & if\ u < 0.5 \\ 1 - [2(1-u)]^{\frac{1}{\eta+1}}, & if\ u \geq 0.5 \end{cases} \tag{2}$$

$$c = b + \delta \Delta_{max} \qquad \delta \in (-1,1) \tag{3}$$

Where $\mathcal{P}(\delta)$ is the polynomial probability distribution, δ is the perturbance factor, *u* is a random number in the range (0,1), *c* is the mutant coral, *b* is the brooder, and Δ_{max} is the maximum permissible range for decision variable. The polynomial distribution produces neighbors by keeping its mean at the current value and its variance as a function of the distribution index η. A large value of η makes brooding more exploitative, and a small value of η makes it more explorative [11].

2.3. Multi-reservoir benchmark problem

Larson first formulated the hypothetical continuous four-reservoir (CFr) problem in 1968 [24] and later adopted by Heidari in 1971 [22]. The current benchmark problem is the one modified by Chow and Cortes-Rivera (1974) [10] and resolved using differential dynamic programming by Murray and Yakowitz (1979) [28].

As shown in Figure 3, the water resources system consists of four reservoirs controlling the flow in two streams, primarily for hydropower production and irrigation purposes. The reservoirs' operation is subject to seasonal storage requirements for flood control, dictating the reservoirs' maximum storage capacity and recreation and fish conservation requirements, imposing a minimum storage capacity. The objective is to maximize the total benefit (*B*) from the system over 12 months of operation periods defined as follows:

$$Maximize\ B = \sum_{r=1}^{R}\sum_{t=1}^{T} b_r(t)\ Re_r(t) \tag{4}$$

Where R is the total number of reservoirs, T is the total number of periods, $b_r(t)$ is the benefit function in period *t* for *r*th reservoir and $Re_r(t)$ is releases in period *t* from reservoir *r*. The fundamental constraints include the continuity equation for each reservoir over each operating period *t* and defined as:

$$S_r(t+1) = S_r(t) + Q_r(t) + RCM_{R\times R}\ Re_r(t) \tag{5}$$

$$RCM_{4\times 4} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix} \tag{6}$$

Where $S_r(t)$ and $S_r(t+1)$ are storages in reservoir *r* at time *t* and *t+1* respectively and $Q_r(t)$ is the inflow volume at period *t* into reservoir *r*. Constraints on releases from the reservoirs and constraints on reservoir storage are:

$$Re_r^{min}(t) \le Re_r(t) \le Re_r^{max}(t) \qquad r = 1, \dots, R\ ; t = 1, \dots, T \tag{7}$$

$$S_r^{min}(t) \le S_r(t) \le S_r^{max}(t) \qquad r = 1, \dots, R\ ; t = 1, \dots, T \tag{8}$$

$$S_r(1) = \ S_r^{initial} \qquad r = 1, \dots, R \tag{9}$$

$$S_r(T+1) = \ S_r^{target} \qquad r = 1, \dots, R \tag{10}$$

Where $Re_r^{min}(t)$, $Re_r^{max}(t)$, $S_r^{min}(t)$, and $S_r^{max}(t)$ are the minimum and the maximum release from/storage of reservoir r at time t, respectively. $S_r^{initial}$ and S_r^{target} are the initial/target volume of the r^{th} reservoir at the beginning of the operation period and the end of the operation period, respectively.

Table 2: The monthly net inflows into reservoirs, maximum allowable reservoir storages, and benefits data of the CFr problem

Data	**Reservoir**	**Period**											
		1	2	3	4	5	6	7	8	9	10	11	12
Net inflows into reservoirs (unit)	1	0.5	1	2	3	3.5	2.5	2	1.25	1.25	0.75	1.75	1
	2	0.4	0.7	2	2	4	3.5	3	2.5	1.3	1.2	1	0.7
	3	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0
Maximum allowable reservoirs storages (unit)	1	-	12	12	10	9	8	8	9	10	10	12	12
	2	-	15	15	15	12	12	12	15	17	18	18	18
	3	-	8	8	8	8	8	8	8	8	8	8	8
	4	-	15	15	15	15	15	15	15	15	15	15	15
Benefits (unit)	1^{Hp*}	1.1	1	1	1.2	1.8	2.5	2.2	2	1.8	2.2	1.8	1.4
	2^{Hp}	1.4	1.1	1	1	1.2	1.8	2.5	2.2	2	1.8	2.2	1.8
	3^{Hp}	1	1	1.2	1.8	2.5	2.2	2	1.8	2.2	1.8	1.4	1.1
	$4^{Hp\ \&\ Irr**}$	2.6	2.9	3.6	4.4	4.2	4	3.8	4.1	3.6	3.1	2.7	2.5

*** Hydropower, ** Irrigation**

Table 3: Constraint parameters ranges for the CFr problem

Reservoir	Minimum storage (unit)	Initial storage (unit)	Target storage (unit)	Minimum release (unit)	Maximum release (unit)
	$\mathbf{S_{min}}$	$\mathbf{S_{initial}}$	$\mathbf{S_{target}}$	$\mathbf{Re_{min}}$	$\mathbf{Re_{max}}$
1	1	6	6	0.005	4
2	1	6	6	0.005	4.5
3	1	6	6	0.005	4.5
4	1	8	8	0.005	8

The reservoirs connectivity matrix (RCM) describes how releases from upstream reservoirs accrue to downstream ones. Data required for modeling the system, such as inflows, reservoir storages, and constraints posed on the CFr problem, are presented in Table 2 and Table 3. Further details for this problem can be found in Murray and Yakowitz (1979) [28]. Since the reported value for global optimum varies in some literature [1,8,16,20], a linear optimization tool (with several release versions) via a couple of different coding approaches (Appendix A) has been applied to ensure that the coding syntax and the tool version do not affect the result when looking for the right global optimum point.

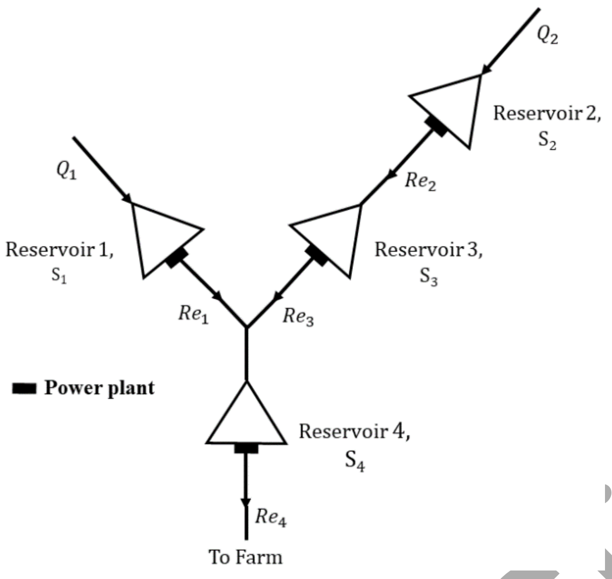


Figure 3: Schematic of the four-reservoir system

2.3.1. Optimization model according to the problem variables

Formulation of the reservoir operation, as an optimization problem, requires the selection of decision variables. Two different sets of decision variables can be considered in reservoir operation problems: storage volumes (S) and release volumes (Re) at each period. These variables are bounded by the recursive continuity equation (5). In a recursive formula, each term is defined as a function of its preceding term(s). The process of recursion can be thought of as climbing a ladder. To get to each rung on the ladder, you must step on the previous rung, whether climbing up or down. Similarly, the two decision variables are both subject to box constraints defined by Eqs. (7)-(10). Therefore, Eq. (5) could be altered in terms of each decision variable, with respect to Eqs. (9) and (10), starting with the first time period or the last one. However, the selection of storage volumes as decision variables leads to easy enforcement of predefined storage volumes at the beginning and at the end of the operation period. By considering release volumes as decision variables, the constrained optimization approach performs much better due to the variable range reduction [1]. Therefore, the number of decision variables would be $R \times T$ as far as the release volumes are considered. The decision variables' sequence to construct the trial solutions will be an R by T matrix as $[Re_1, Re_2, \ldots, Re_T]^R$.

According to the search domain, two approaches have been studied to solve constrained optimization problems. The first is searching in a broader domain by applying a penalty method as the most common approach. This could be done by taking some of the constraints into account via adding a term to the objective function that prescribes a relatively high cost for violation constraints. In this approach, the search algorithm may offer not acceptable solutions in terms of violating constraints. However, the feasible direction method (primal direction method) works on the original problem by considering all restrictions via searching the feasible region for an optimal solution. Therefore, the search domain would be as narrow as possible. Formulating the problem and setting up the algorithm to deal with the feasible space is challenging, especially in non-convex optimization problems, like the one at hand.

2.3.2. CFr operation using CRO algorithm with the penalty method

All initial corals in the reef would be generated in the interval $[Re_r^{min}, Re_r^{max}]$ satisfying the corresponding constraint in Eq. (7) when using the penalty approach. During sexual reproduction, this range is similarly used to limit the reproduced larvae. For this purpose, each larva enters a procedure named *larvae correction* to apply the associated limitation by projecting the outbound solutions on edge. For those temporary solutions that violate the state variable's constraints defined by Eqs. (8) and (10), a lower benefit allocates by the operation policy. This can be achieved by a penalty method in which the operation's total cost is considered as the sum of the operation cost defined by Eq. (4) and a penalty cost defined as:

$$B' = \sum_{r=1}^{R}\sum_{t=1}^{T}\sum_{i=1}^{3} b_r(t)\, Re_r(t) - P_t^i(r) \tag{11}$$

$$P_t^i(r) = g \times V_t^i(r), \qquad \forall i \in \{1,2,3\} \tag{12}$$

$$V_t^1(r) = \begin{cases} S_r^{min}(t) - S_r(t) & ;\ if\ \ S_r(t) < S_r^{min}(t) \\ 0 & ;\ Otherwise \end{cases} \tag{13}$$

$$V_t^2(r) = \begin{cases} S_r(t) - S_r^{max}(t) & ; \, if \;\; S_r^{max}(t) < S_r(t) \\ 0 & ; \, Otherwise \end{cases} \qquad (14)$$

$$V_t^3(r) = \begin{cases} \left|S_r(T+1) - S_r^{target}\right| & ; \, if \;\; S_r(T+1) \neq S_r^{target} \\ 0 & ; \, Otherwise \end{cases} \qquad (15)$$

Where, B' is the total penalized benefit function, $P_t^i(r)$ is a penalty function related to constraints in Eqs. (8) and (10), $V_t^i(r)$ is the violation function and, g represents the penalty parameter with a large enough positive value to force the search into a feasible region. It doesn't worth mentioning that the penalty term in Eq. (11) will be zero for feasible solutions.

2.3.3. CFr operation using constrained CRO algorithm

Search in a set of feasible solutions has been used in the PSO algorithm [1,2] by adapting velocity vectors, which makes keeping solutions inside the feasible-space easy. In contrast, the CRO algorithm belongs to a class of search methods (like genetic algorithms) in which the offspring solutions are constructed incrementally by recombination of old solutions. This can be a challenge for optimization problems with recursive constraints since constraints' violation is very likely; hence, newly generated larvae should be modified based on all the constraint equations. For this purpose, a new set of bounds is constructed to initialize the reef considering the continuity Eq. (5) and storage volume constraints (8) at the end of the period *t*. Replacing $S_r(t+1)$ from Eq. (5) in Eq. (8) leads to the following constraints for the release amount, assuming a known value of $S_r(t)$ for storage volume at the beginning of period *t*:

$$S_r(t) + Q_r(t) - S_r^{max}(t+1) \leq Re_r(t) \leq S_r(t) + Q_r(t) - S_r^{min}(t+1) \qquad (16)$$

Combining Eq. (16) with the original box constraint of (7) leads to the following constraints for the release volumes for period *t*.

$$Max\left(S_r(t) + Q_r(t) - S_r^{max}(t+1)\, , Re_r^{min}(t)\,\right) \;\leq\; Re_r(t) \;\leq \qquad (17)$$

$$Min\left(S_r(t)+Q_r(t)-S_r^{min}(t+1)\,,Re_r^{max}(t)\right)$$

At the larvae correction step, the offspring corals should be examined for breached rates after reproductive operators have been applied through a local search. The treatment should be based on the bits of the solution string that caused the violation. For instance, if the last decision variable, *Re*$Re_r(12)$, did not preserve Eq. (10) (i.e., a violation in the final storage amount), the deviation will be distributed on previous variables due to their permitted space. In the case of violations by an intermediate variable, the same process applies to antecedent variables. While letting the solutions thrive in the feasible space, these operators provide a proper local search (exploitation).

This approach could achieve better solutions and improved convergence characteristics by reducing the search space size. Furthermore, it eliminates the problem of tuning the penalty parameter for better method's performance. Since the *Larvae Correction* operator always produces permissible solutions, the constrained CRO's search effort to converge to an optimal or a near-optimal solution is expected to be much smaller than other approaches where any value is allowed during the optimization process.

2.4. Improving the CCRO algorithm using the Q-Learning method (CCRO-QL)

Reinforcement learning is an area of machine learning focused on how learning agents should take actions in an environment to achieve a goal, such as maximizing cumulative reward. The agent, during its course of learning, experiences various situations in the environment (*States*). The agent in each state may choose an allowable action that may get a reward (or penalty). Over time, the agent learns to maximize these rewards to behave optimally at any given state. In problems where a model of the environment is not available or cannot be learned, model-free methods as an explicit trial-and-error algorithm should be used. "Q-learning is a model-free reinforcement learning algorithm to learn the quality of actions telling an agent what action to take under what circumstances" [42]. Q-Learning uses Q-values (also called Action-Values) to improve the behavior of the learning agent iteratively. Q-Values, $Q(\mathcal{S},A)$, is an estimation of how good it is to take action *A* at the state

$\mathcal{S}$. This estimation will be iteratively computed using the *Temporal Difference (TD) Update* rule. The TD-Update rule, which is applied at every time step of the agents' interaction with the environment, can be represented as follows:

$$Q(\mathcal{S}, A) \leftarrow Q(\mathcal{S}, A) + \alpha(\mathcal{R} + \gamma \max_{A'} Q(\mathcal{S}', A') - Q(\mathcal{S}, A)) \tag{18}$$

Where $\mathcal{S}$ is the agent's current state, and A is the current action picked according to some policy. $\mathcal{S}'$ is the next state where the agent ends up, A' is the next best action to be selected using current Q-value estimation, and $\mathcal{R}$ is the current reward observed from the environment. γ is a parameter, $0 \leq \gamma \leq 1$, called the discount rate that determines the present value of the future reward. α is a positive constant step-size parameter that influences the rate of learning.

As was mentioned in section 2.3.3, the recombination of old solutions to build offspring solutions is not an effective strategy in recursive optimization problems since the constraints' violation is common. If the operator used for the external sexual reproduction (ESR) is not carefully selected, the rate of change in larvae as a result of the larvae correction phase will be such that larvae will not benefit from their broadcast spawners, as if the crossover has led to a random offspring. Under such conditions, the exploitation part of the algorithm will be lost. As a solution to this issue, the authors proposed a hybrid method to make the exploitation phase more efficient using a machine learning approach. The authors believe that eliminating the ESR operator alongside upgrading the ISR operator using the Q-Learning method can improve the CRO algorithm's exploitation efficiency. The aim is to select some bits purposefully in a brooder to participate in the ISR process instead of random selection. The brooding operator remains a polynomial probability distribution (Eq. (1)). Since the broadcast spawning operator is removed, all corals in the reef will enter the brooding stage in every iteration.

Assume that each solution for the problem (i.e., water release amount) is a state ($\mathcal{S}$). The possible actions are sets of some nominated bits out of a coral's bits (4×12=48). For each coral, Q-Values is another 4×12 matrix that indicates the importance of each bit in the solution's matrix. The benefit rate in Table S.2 would be a proper beginning action-values after the reef initialization. The agent is reward-motivated and will learn how to pick bits through feedback from the environment; hence, certain rewards (including penalties) and their magnitude are needed accordingly. The agent will receive a positive reward if the objective function is improved, ΔB, (i.e., total benefit in Eq.(4)); otherwise, it will be penalized for decreasing the total benefit. Accordingly, the reward can be defined as the normalized ratio of ΔB to the bit's shift amount. Also, it should be noted that a high value for the discount factor (γ) captures the effective long-term award, whereas a low discount factor makes the agent consider an immediate reward.

The tradeoff between exploration and exploitation is controlled by a policy called ε, preventing possible overfitting during training. In an ε-greedy policy, most of the time, the algorithm chooses an action with maximal estimated action-values. Yet with a small probability (ε), the action might be selected randomly. The values of α, γ, and ε are mostly based on intuition and some based on hit and trial. Generally, all three should decrease over time because it builds up more resilient priors as the agent continues to learn. Figure 4 illustrates the pseudo-code of the CCRO-QL algorithm with different phases.

Due to the interaction between the learning agent and its environment in terms of states, actions, and rewards, the CCRO-QL method uses the formal framework of Markov decision processes (MDP). This framework represents the artificial intelligence problem's essential features, such as a sense of cause and effect, uncertainty and nondeterminism, and an explicit goal [42]. In CCRO-QL, all agents' population learn according to a common policy; therefore, it can be considered multi-agent reinforcement learning.

3. Hyperparameters optimization

The conventional way to achieve hyperparameter optimization is to perform a grid search or a parameter sweep; an exhaustive search through a manually designated subset of the feasible space. In grid search, the set of trials is formed by assembling a discrete combination of possible values. In both approaches (CRO and CCRO), a two-stage grid search has been performed for the model operators and their parameters. In the first stage, based on the CRO evaluation of mathematical test functions, three selection methods (i.e., roulette wheel selection, tournament selection, and uniform random selection) are used to choose broadcast spawners. Two conventional methods, geometric and arithmetic crossover operators, were investigated as well. Three well-known probability distributions, the Gaussian, Cauchy, and polynomial distributions, were designated for the brooding operator. In the second stage, a search through a fine grid was performed for all parameters mentioned in Table S.1. Another full grid search under the CCRO-QL model was used to optimize the step-size and discount-rate parameters, applying the same principles. Table S.4 displays allowable ranges and the best values chosen for each parameter.

4. Results and discussions

4.1. CRO evaluation in mathematical test functions

Table 4 displays the statistical performance metrics achieved by two different CRO versions (with Polynomial and Gaussian-Cauchy internal reproduction, respectively) in the benchmark functions. It also includes the results obtained with the PSO algorithm [40]. It is noteworthy that the polynomial mutation-CRO$^{(P)}$ version can obtain better results than the others, while the Kruskal-Wallis test shows positive statistical significance in all CBP functions. This observation demonstrates that the CRO algorithm offers a satisfactory performance level in continuous optimization problems, which supports the results presented by Li et al. [25], Yang et al. [49], Salcedo-Sanz et al. [38–40]. Figure 5 shows the evolution of PSO solutions and the two CRO versions for two functions out of the five CBP.

Table 4: Comparison of obtained results (mean ± standard deviation) from PSO and CRO in different CBP functions of the 30 runs

Algorithm	Rosenbrock	Schwefel 2.22	Sphere	Rastrigin	Quartic
PSO	$2.27 \cdot 10^{-6} \pm 2 \cdot 10^{-6}$	$2.1 \cdot 10^{-4} \pm 7 \cdot 10^{-4}$	$1.24 \cdot 10^{-3} \pm 3 \cdot 10^{-3}$	$4.719 \cdot 10^{-3} \pm 2.32 \cdot 10^{-3}$	$2 \cdot 10^{-2} \pm 1 \cdot 10^{-2}$
CRO$^{(G+C)}$	$2.29 \cdot 10^{-6} \pm 1 \cdot 10^{-6}$	$1.83 \cdot 10^{-3} \pm 3 \cdot 10^{-4}$	$1.3 \cdot 10^{-3} \pm 2 \cdot 10^{-4}$	$2.27 \cdot 10^{-6} \pm 2 \cdot 10^{-6}$	$2 \cdot 10^{-2} \pm 9 \cdot 10^{-3}$
CRO$^{(P)}$	$1.55 \cdot 10^{-6} \pm 4.2 \cdot 10^{-6}$	$4.3 \cdot 10^{-142} \pm 1 \cdot 10^{-141}$	$6.8 \cdot 10^{-132} \pm 3 \cdot 10^{-131}$	0 ± 0	$1.4 \cdot 10^{-4} \pm 1.3 \cdot 10^{-4}$
Stat. Sig.	+	+	+	+	+

4.2. CRO evaluations in the CFr problem

Except for the reef size, the CRO parameters and operators' values are initialized based on the previous step results, as shown in Table S.1. Since the number of decision variables is *n=48*, the initial population is assumed to be about *n×10*. Therefore, the artificial coral reef initially consists of a square grid of *22×22* cells, depicting the maximum number of solutions available for evaluation per generation. The objective function's optimal value (B) obtained from linear programming implemented by the *Lingo 12.0, 14.0, 17.0, 18.0* [52] (linear optimization tool) equals 308.2915 for all versions, and coding syntax illustrated in Appendix A. This is the same value reported in [20], and it could be assumed as the final global optimum. It doesn't worth mentioning that the obtained solutions (Re) from all executions were identical as well.

Almost all recent studies that applied meta-heuristic methods to solve the CFr benchmark problem are listed in Table S.6. This table summarizes the proposed methods, the number of function evaluations, the average, and the best results obtained in each study. The exact values of the optimal decision variables (i.e., monthly release volumes of each reservoir) are not reported in the cited research. According to the presented diagrams, there are differences in values in some instances that may be due to the presence of several optimal solutions or the provided numbers' decimal accuracy, while no clear evidence to distinct these states has been provided yet.

4.2.1. CCRO in comparison to the original CRO

Initial corals produced solely by considering Eq. (7) in the CRO, while in the constrained CRO (CCRO), all Eqs. (7)-(10) should be met. Both forward and backward methods can do modification of the generated larvae. In the forward method, with the primary assumption $S_r(1) = \ S_r^{initial}$ in each reservoir, the answers are generated using the continuity equation and constraints (7) and (8). In the backward method, these steps are inverted with the initial assumption $S_r(T+1) = \ S_r^{target}$ and rewriting Eq. (5) in terms of $S_r(t)$. To keep the diversity, both forward and backward approaches were used based on a generated uniform random number. Table S.4 displays the minimum, maximum, and step size for each CRO control parameter.

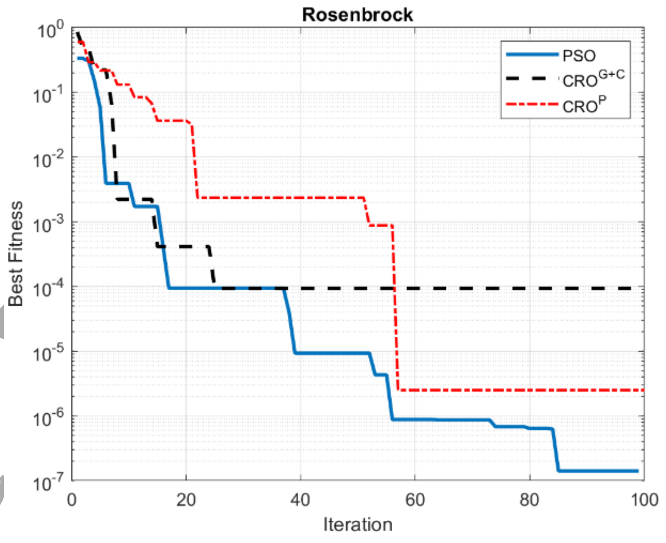
The best values obtained through hyperparameter optimization have been used for all CRO versions solving the CFr problem. The number of function evaluations was 300,000 in each run, gained by trial-and-error to be competitive with the lowest NFE in the literature review and to reach the correct global optimum.

Algorithm 1: Pseudo-code for the CCRO-QL algorithm

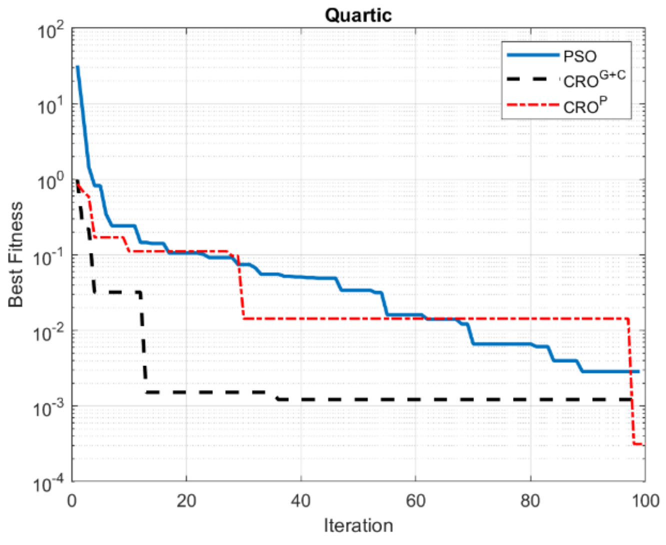
```
Function CCRO-QL(Problem) returns a state that is a local optimum
Input: M, N, $\rho_o$, $\kappa$, $F_d$, $F_a$, $P_d$, $\alpha$, $\gamma$, $\varepsilon$, ProblemSize
Output: $S_{best}$ ($Re_{best}$)

1:  REEF ← InitializePopulation(M, N, $\rho_o$, ProblemSize);
2:  $Q(\mathcal{S},A)$ ← Initialize $Q(\mathcal{S},A)$;
3:  $\mathcal{R} \leftarrow \emptyset$;
4:  EvaluatePopulation(REEF);
5:  $S_{best}$ ← GetBestSolution(REEF);
6:  while TerminalCondition() do
7:      foreach Coral ∊ REEF do
8:          Larvae ← Brooding(Coral, $Q(\mathcal{S},A)$, $\mathcal{E}$);
9:          Larvae ← LarvaeCorrection(Larva);
10:     end
11:     foreach Larva do
12:         EvaluatePopulation(Larva);
13:         $\mathcal{R}$ ← EvalFitnessImprovement;
14:         $Q(\mathcal{S},A) \leftarrow Q(\mathcal{S},A) + \alpha(\mathcal{R} + \gamma \max_{A'} Q(\mathcal{S}',A') - Q(\mathcal{S},A))$;
15:     end
16:     REEF ← LarvaeSetting(REEF, Larvae, $\mathcal{K}$);
17:     ALarvae ← Budding(REEF, $F_a$);
18:     REEF ← LarvaeSetting(REEF, ALarvae, $\mathcal{K}$);
19:     REEF ← Depredation(REEF, $F_d$, $P_d$);
20:     $S_{best}$ ← GetBestSolution(REEF);
21: end
22: return $S_{best}$;
```

Figure 4: Pseudo-code for the CCRO-QL algorithm



(a)



(b)

Figure 5: Evolution of a) Rosenbrock and b) Quartic objective functions obtained by the PSO, CRO^{G+C}, and CRO^{P}

Table 5. Performance comparison of the proposed methods for the four-reservoir problem

Model	**Total benefit**			
Run #	CRO	CCRO	CCRO-QL	LP (Global solution)
1	301.77	307.38	308.2886	308.2915
2	303.49	307.16	308.2216	
3	302.40	307.42	308.2533	
4	303.93	307.63	308.2879	
5	302.98	307.14	308.2576	
6	301.78	307.41	308.2858	
7	304.71	307.40	308.2889	
8	301.51	307.10	308.2906	
9	302.85	307.24	308.2886	
10	301.38	307.27	308.2899	
Best	304.71	307.63	308.2906	
Mean	302.68	307.31	308.275	
STD	1.118	0.162	0.023	
p-value	-	1.4 e -10	3.4 e -13	

Table 5 presents the statistical results obtained by all CRO versions (i.e., the original CRO algorithm, CCRO, and CCRO-QL) for ten different algorithms' independent runs. The best and average benefit show that the CCRO results are much better than those of the original CRO. The CRO and CCRO converged to the best objective function values of 304.71 and 307.63, respectively.

From Table 5, it can be deduced that CCRO is more robust than the original CRO because its STD value is seven times smaller than that of the CRO, which implies that the algorithm uncertainty decreased by taking the primal direction method. The p-value obtained from Student's T-Test rejects the null hypothesis. Thus, there is a significant difference between the CRO results and the CCRO's at the 99% confidence level.

4.2.2. CCRO-QL Performance

Statistics, including mean, standard deviation, and p-value in Table 5, depict that CCRO-QL outperforms other proposed methods. The agent learns to pick more effective bits in each step properly. In multi-agent reinforcement learning, the scenario in which all the agents try to maximize their own reward signal that they receive is known as a competitive setting. In this situation, conflicts of interest among the agents are likely, implying that some agents' proper actions are improper for others. That is why fixed hyperparameters (i.e., step-size, discount-rate, and ε probability) did better than adaptive ones in terms of converging to local minimums for the CFr problem.

The best objective function obtained with the CCRO-QL method equaled 308.29 at two decimal accuracy, which is the same as the maximum global value achieved via linear programming (LP). Since the precise global optimum is guaranteed through linear programming, decimal accuracy is necessary to distinguish the local optimum. Figure 6 displays three different paths with an equal total benefit (B) at two decimal place accuracy, although two do not correspond to the global optimum. Figure 7 a and b compares the optimal release volume and storage volume, obtained using the CCRO-QL algorithm (with B =308.2906 and NFE=300,000) and those of the LP method. The fact that the constrained algorithm is a search method in the feasible region of the search space can be best illustrated by the convergence of the solutions shown in Figure 8. It is evident that both CCRO and CCRO-QL algorithms start with B above 275, while the CRO algorithm could produce this level of solution later in the search. The maximum and average results, respectively, with NFE≈100,000 were 308.2628 and 308.0355, which is still reliable. The proposed method could obtain the exact global optimum (i.e., 308.2915) by performing 880,000 function evaluations. Table S.5 shows the release amount obtained by LP and CCRO-QL for the CFr problem, which has never been provided in the literature. As seen from the table, all decision variable values converge to those from LP. Therefore, it can be concluded that the CFR problem only has one global optimum (i.e., unimodal function), which is the same as a result obtained via LP. It should be noted that almost all literature results (Table S.6) as optimal have nearly the same decision-variable values according to the best fitness with two decimal places of accuracy. Yet, the results better than LP would be questionable since linear programming produces the final global optimum as a deterministic numerical method.

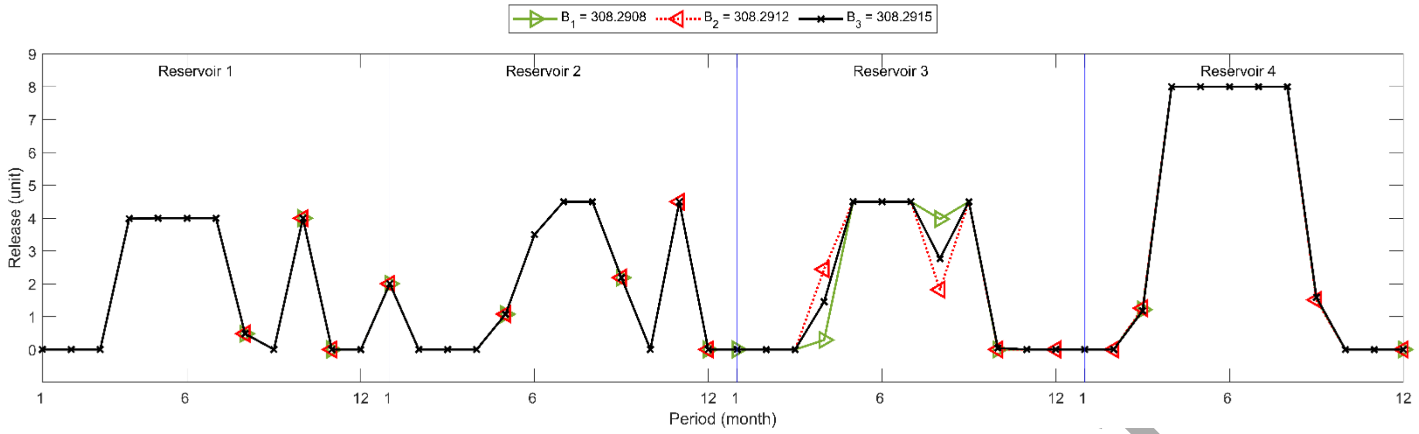


Figure 6: Three unequal solutions obtained from the CCRO-QL

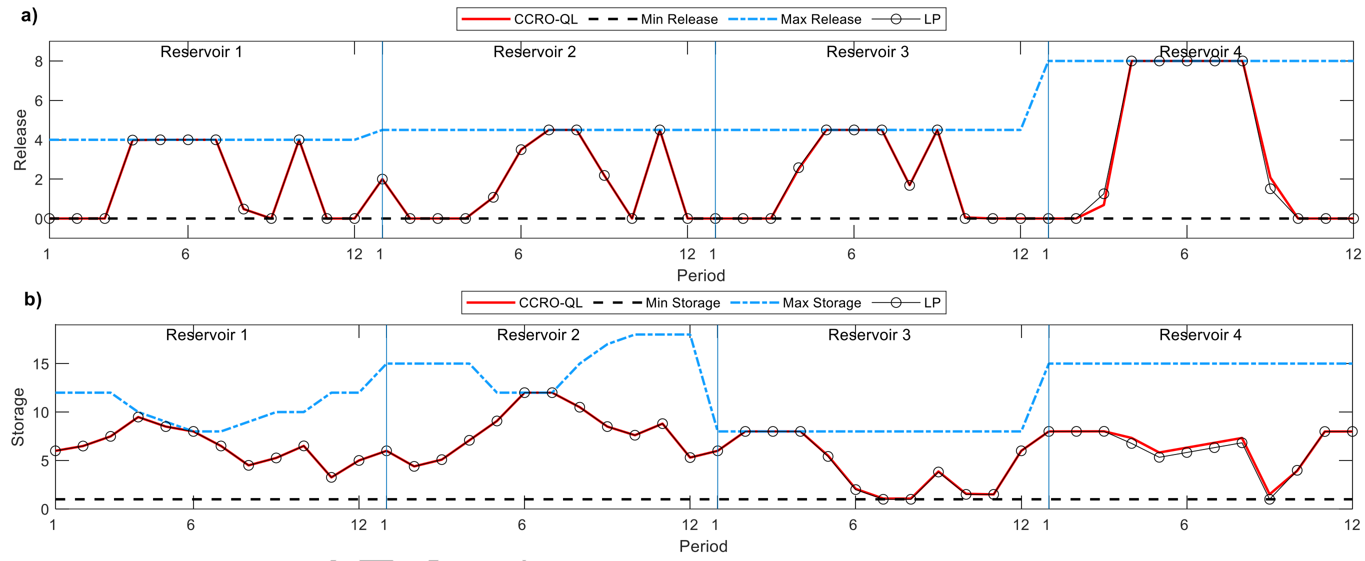


Figure 7: Comparison of the a) release volumes and b) storage volumes obtained using the CCRO-QL algorithm with the LP method

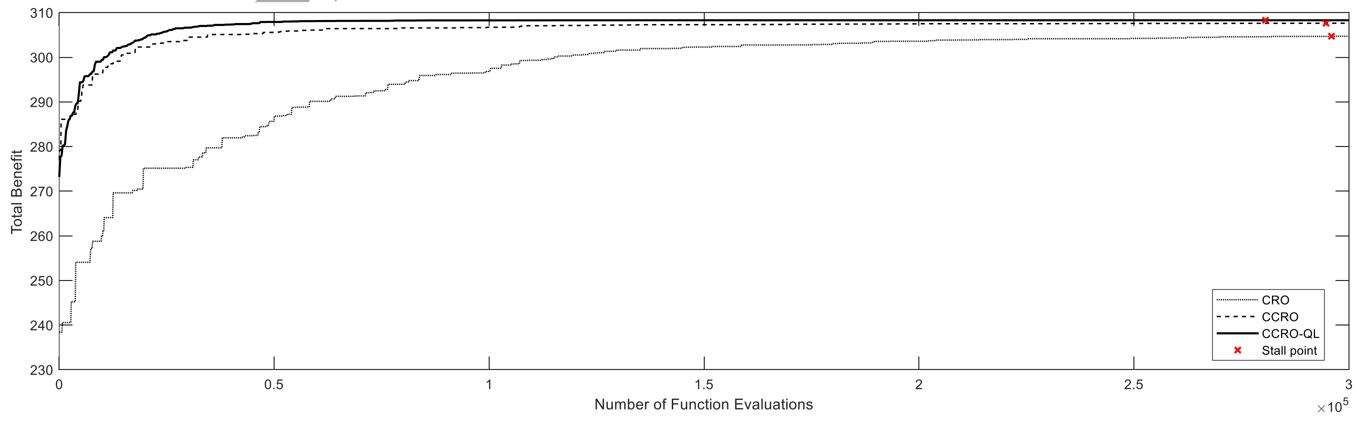


Figure 8: Convergence curves of the total benefit for the CRO, CCRO, and CCRO-QL algorithm

5. Conclusions

This paper has solved the continuous four-reservoir problem as a benchmark problem in surface water resources management. Three versions of a meta-heuristic algorithm have been used to tackle the problem, the so-called Coral Reefs Optimization algorithm that simulates reproduction and formation of coral reefs. The first step has shown that the proposed approach can deal with continuous problems through experiments carried out for five well-known benchmark functions. The CRO has obtained reliable and robust solutions, which is supported by previous research [39,40]. According to the problem's characteristics, two modifications have been proposed to the original CRO. First, search in the feasible region instead of the entire solution domain to increase the calculation-cost efficiency of solving recursive functions. The applied search methodology is the primal direction method, which ensures the results' consistency by producing possible solutions regarding the problem's constraints. Thereafter, the previous approach was hybridized with the Q-Learning method to improve reproduction functions. This method outperformed other problem-solving heuristics in terms of the precision of the obtained solution and the number of objective function evaluations. The CCRO-QL algorithm showed excellent performance by yielding a solution equal to the global optimum. Overall, those results indicate that the optimized solutions' uncertainty decreases after narrowing the search space and modifying the reproduction operators because the uncertainty was mainly caused by penalized offspring and an inefficient recombination method.

6. Acknowledgments

The authors would like to thank Alireza Kazemi (Center for Mind and Brain, Department of Psychology, University of California, Davis, CA, USA) for his valuable suggestions in setting up the Q-Learning method. The authors are grateful to the Editor-in-Chief Jason M. Evans and anonymous reviewers for their careful review and helpful comments that improved the original paper.

7. Supplementary Materials

The following are available online at the Journal of Environmental Management website: Table S.1: Control parameters' values used in the CRO algorithm to solve CBP, Table S.2: Monthly net inflows into reservoirs, maximum allowable reservoir storages, and benefits data of the CFr problem, Table S.3: Constraint parameters' ranges for the CFr problem, Table S.4: Control parameters' values used in the CRO algorithm to solve CFr problem, Table S.5: Value of decision variables obtained by the LP method and CCRO-QL algorithm, Table S.6: Comparison of reported benefits by evolutionary approaches for the continuous four-reservoir problem, and two *Lingo* sample codes for Continuous Four-reservoir Problem.

References

1. Afshar MH. *Extension of the Constrained Particle Swarm Optimization Algorithm to Optimal Operation of Multi-Reservoirs System*. Int J Electr Power Energy Syst [Internet]. 2013;51:71–81. Available from: http://www.sciencedirect.com/science/article/pii/S0142061513000951

2. Afshar MH. *Large Scale Reservoir Operation by Constrained Particle Swarm Optimization Algorithms*. J Hydro-environment Res [Internet]. 2012;6(1):75–87. Available from: http://www.sciencedirect.com/science/article/pii/S1570644311000979

3. Agrawal S, Panda R, Abraham A. *A Novel Diagonal Class Entropy-Based Multilevel Image Thresholding Using Coral Reef Optimization*. IEEE Trans Syst Man, Cybern Syst. 2018;1–9.

4. Ahmadi A, Nasseri M. *Do direct and inverse uncertainty assessment methods present the same results?* J Hydroinformatics [Internet]. 2020 Jul 1;22(4):842–55. Available from: https://iwaponline.com/jh/article/22/4/842/74130/Do-direct-and-inverse-uncertainty-assessment

5. Ahmadi A, Nasseri M, Solomatine DP. *Parametric uncertainty assessment of hydrological models: coupling UNEEC-P and a fuzzy general regression neural network*. Hydrol Sci J [Internet]. 2019 Jul 4;64(9):1080–94. Available from: https://doi.org/10.1080/02626667.2019.1610565

6. Azgomi H, Sohrabi MK. *A Novel Coral Reefs Optimization Algorithm for Materialized View Selection in Data Warehouse Environments*. Appl Intell [Internet]. 2019; Available from: https://doi.org/10.1007/s10489-019-01481-w

7. Azizipour M, Sattari A, Afshar MH, Goharian E, Solis SS. *Optimal hydropower operation of multi-reservoir systems: hybrid cellular automata-simulated annealing approach*. J Hydroinformatics. 2020;22(5):1236–57.

8. Bozorg-Haddad O, Afshar A, Mariño MA. *Multireservoir Optimisation in Discrete and Continuous Domains*. In: Proceedings of the Institution of Civil Engineers-Water Management. Thomas Telford Ltd; 2011. p. 57–72.

9. Camacho-Gómez C, Marsa-Maestre I, Gimenez-Guzman JM, Salcedo-Sanz S. *A Coral Reefs Optimization Algorithm with Substrate Layer for Robust Wi-Fi Channel Assignment*. Soft Comput. 2019;

10. Chow V Te, Cortes-Rivera G. *Application of Dddp in Water Resources Planning.* Ill Univ Water Resour Cent Res Rep. University of Illinois at Urbana-Champaign. Water Resources Center; 1974.

11. Deb K, Agrawal RB. *Simulated Binary Crossover for Continuous Search Space*. Complex Syst [Internet]. 1994;9(2):1–34. Available from: citeulike-article-id:2815748%5Cnhttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.26.8485

12. Deb K, Goyal M. *A Combined Genetic Adaptive Search (GeneAS) for Engineering Design*. Comput Sci Informatics. 1996;26:30–45.

13. Dodangeh E, Panahi M, Rezaie F, Lee S, Tien Bui D, Lee CW, et al. *Novel hybrid intelligence models for flood-susceptibility prediction: Meta optimization of the GMDH and SVR models with the genetic algorithm and harmony search*. J Hydrol. 2020;590:125423.

14. Durán-Rosal AM, Gutiérrez PA, Salcedo-Sanz S, Hervás-Martínez C. *A Statistically-Driven Coral Reef Optimization Algorithm for Optimal Size Reduction of Time Series*. Appl Soft Comput J. 2018;63:139–53.

15. Durán-Rosal AM, Gutiérrez PA, Salcedo-Sanz S, Hervás-Martínez C. *Dynamical Memetization in Coral Reef Optimization Algorithms for Optimal Time Series Approximation*. Prog Artif Intell. 2019;8(2):253–62.

16. Ehteram M, Allawi MF, Karami H, Mousavi S-F, Emami M, EL-Shafie A, et al. *Optimization of Chain-Reservoirs' Operation with a New Approach in Artificial Intelligence*. Water Resour Manag [Internet]. 2017 May 4;31(7):2085–104. Available from: https://doi.org/10.1007/s11269-017-1625-6

17. Ehteram M, Mousavi S-F, Karami H, Farzin S, Emami M, Binti Othman F, et al. *Fast convergence optimization model for single and multi-purposes reservoirs using hybrid algorithm*. Adv Eng Informatics [Internet]. 2017 Apr;32:287–98. Available from: https://doi.org/10.1016/j.aei.2017.04.001

18. Ficco M, Esposito C, Palmieri F, Castiglione A. *A Coral-Reefs and Game Theory-based Approach for Optimizing Elastic Cloud Resource Allocation*. Futur Gener Comput Syst. 2018;78:343–52.

19. Garcia-Hernandez L, Salas-Morera L, Garcia-Hernandez JA, Salcedo-Sanz S, Valente de Oliveira J. *Applying the Coral Reefs Optimization Algorithm for*

Solving Unequal Area Facility Layout Problems. Expert Syst Appl [Internet]. 2019;138:112819. Available from: http://www.sciencedirect.com/science/article/pii/S0957417419305214

20. Garousi-Nejad I, Bozorg-Haddad O, Loáiciga HA. *Modified Firefly Algorithm for Solving Multireservoir Operation in Continuous and Discrete Domains*. J Water Resour Plan Manag [Internet]. 2016;142(9):4016029. Available from: https://doi.org/10.1061/(ASCE)WR.1943-5452.0000644

21. Gomez J. *Self Adaptation of Operator Rates in Evolutionary Algorithms*. In: Deb K, editor. Genetic and Evolutionary Computation – GECCO 2004. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004. p. 1162–73.

22. Heidari M, Chow V Te, Kokotović P V, Meredith DD. *Discrete Differential Dynamic Programing Approach to Water Resources Systems Optimization*. Water Resour Res. 1971;7(2):273–82.

23. Jiménez-Fernández S, Camacho-Gómez C, Mallol-Poyato R, Fernández J, Del Ser J, Portilla-Figueras A, et al. *Optimal Microgrid Topology Design and Siting of Distributed Generation Sources Using a Multi-Objective Substrate Layer Coral Reefs Optimization Algorithm*. Sustainability. 2019;11(1):169.

24. Larson RE. *State Increment Dynamic Programming*. 1968;

25. Li M, Miao C, Leung C. *A Coral Reef Algorithm Based on Learning Automata for the Coverage Control Problem of Heterogeneous Directional Sensor Networks*. Sensors. 2015;15(12).

26. Moeini R, Soghrati F. *Optimum outflow determination of the multi-reservoir system using constrained improved artificial bee colony algorithm*. Soft Comput. 2020;24(14):10739–54.

27. Mohammadi M, Farzin S, Mousavi SF, Karami H. *Investigation of a New Hybrid Optimization Algorithm Performance in the Optimal Operation of Multi-Reservoir Benchmark Systems*. Water Resour Manag. 2019;33(14):4767–82.

28. Murray DM, Yakowitz SJ. *Constrained Differential Dynamic Programming and Its Application to Multireservoir Control*. Water Resour Res. 1979;15(5):1017–27.

29. Razavi Termeh SV, Kornejady A, Pourghasemi HR, Keesstra S. *Flood susceptibility mapping using novel ensembles of adaptive neuro fuzzy inference system and metaheuristic algorithms*. Sci Total Environ. 2018;615:438–51.

30. Salcedo-Sanz S. *A Review on the Coral Reefs Optimization Algorithm: New Development Lines and Current Applications*. Prog Artif Intell. 2017;6(1):1–15.

31. Salcedo-Sanz S, García-Herrera R, Camacho-Gómez C, Alexandre E, Carro-Calvo L, Jaume-Santero F. *Near-optimal Selection of Representative Measuring Points for Robust Temperature Field Reconstruction with the CRO-SL and Analogue Methods*. Glob Planet Change. 2019 Jul 1;178:15–34.

32. Salcedo-Sanz S, Camacho-Gómez C, Magdaleno A, Pereira E, Lorenzana A. *Structures Vibration Control Via Tuned Mass Dampers Using a Co-Evolution Coral Reefs Optimization Algorithm*. J Sound Vib. 2017;

33. Salcedo-Sanz S, Casanova-Mateo C, Pastor-Sánchez A, Sánchez-Girón M. *Daily Global Solar Radiation Prediction Based on a Hybrid Coral Reefs Optimization – Extreme Learning Machine Approach*. Sol Energy. 2014;105:91–8.

34. Salcedo-Sanz S, Gallo-Marazuela D, Pastor-Sánchez A, Carro-Calvo L, Portilla-Figueras A, Prieto L. *Offshore Wind Farm Design with the Coral Reefs Optimization Algorithm*. Renew Energy. 2013;63:109–15.

35. Salcedo-Sanz S, García-Díaz P, Portilla-Figueras JA, Del Ser J, Gil-López S. *A Coral Reefs Optimization Algorithm for Optimal Mobile Network Deployment with Electromagnetic Pollution Control Criterion*. Appl Soft Comput. 2014;24:239–48.

36. Salcedo-Sanz S, Jiménez-Fernández S, Aybar-Ruiz A, Casanova-Mateo C, Sanz-Justo J, García-Herrera R. *A CRO-species Optimization Scheme for Robust Global Solar Radiation Statistical Downscaling*. Renew Energy. 2017;111:63–76.

37. Salcedo-Sanz S, Pastor-Sánchez A, Gallo-Marazuela D, Portilla-Figueras A. *A Novel Coral Reefs Optimization Algorithm for Multi-objective Problems*. In: Yin H, Tang K, Gao Y, Klawonn F, Lee M, Weise T, et al., editors. Intelligent Data Engineering and Automated Learning – IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, October 20-23, 2013 Proceedings. Berlin, Heidelberg: Springer; 2013. p. 326–33.

38. Salcedo-Sanz S, Pastor-Sánchez A, Portilla-Figueras JA, Prieto L. *Effective Multi-Objective Optimization with the Coral Reefs Optimization Algorithm*. Eng Optim. 2016;48(6):966–84.

39. Salcedo-Sanz S, Del Ser J, Gil-López S, Landa-Torres I, Portilla-Figueras JA. *The Coral Reefs Optimization Algorithm: A New Metaheuristic Algorithm for Hard Optimization Problems*. In: Proceedings of the 15th International Conference on Applied Stochastic Models and Data Analysis (ASMDA). Matar´o, Barcelona; 2013. p. 751–8.

40. Salcedo-Sanz S, Del Ser J, Landa-Torres I, Gil-López S, Portilla-Figueras JA. *The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems*. Sci World J. 2014;2014:15.

41. Samadi-koucheksaraee A, Ahmadianfar I, Bozorg-Haddad O, Asghari-pari SA. *Gradient Evolution Optimization Algorithm to Optimize Reservoir Operation Systems*. Water Resour Manag. 2019;33(2):603–25.

42. Sutton RS, Barto AG. *Reinforcement Learning: An introduction*. MIT Press; 2018.

43. Tsai CW, Chang WY, Wang YC, Chen H. *A High-Performance Parallel Coral Reef Optimization for Data Clustering*. Soft Comput [Internet]. 2019;23(19):9327–40. Available from: https://doi.org/10.1007/s00500-019-03950-3

44. Wikipedia contributors. *Metaheuristic* [Internet]. Wikipedia, The Free Encyclopedia. 2020 [cited 2020 Mar 7]. Available from: https://en.wikipedia.org/w/index.php?title=Metaheuristic&oldid=933691575

45. Wikipedia contributors. *Kruskal–Wallis one-way analysis of variance* [Internet]. Wikipedia, The Free Encyclopedia. 2020 [cited 2020 Mar 8]. Available

from: https://en.wikipedia.org/w/index.php?title=Kruskal–Wallis_one-way_analysis_of_variance&oldid=941704728

46. Wolpert DH, Macready WG. *No Free Lunch Theorems for Optimization*. Vol. 1, IEEE Transactions on Evolutionary Computation. Technical Report SFI-TR-95-02-010, Santa Fe Institute; 1997.

47. Wolpert DH, Macready WG. *No Free Lunch Theorems for Search*. Technical Report SFI-TR-95-02-010, Santa Fe Institute; 1995.

48. Yan C, Ma J, Luo H, Patel A. *Hybrid Binary Coral Reefs Optimization Algorithm with Simulated Annealing for Feature Selection in High-Dimensional Biomedical Datasets*. Chemom Intell Lab Syst. 2019;184:102–11.

49. Yang Y, Yang B, Niu M. *Parameter Identification of Jiles–Atherton Model for Magnetostrictive Actuator Using Hybrid Niching Coral Reefs Optimization Algorithm*. Sensors Actuators, A Phys. 2017;261:184–95.

50. Yang Z, Zhang T, Zhang D. *A Novel Algorithm with Differential Evolution and Coral Reef Optimization for Extreme Learning Machine Training*. Cogn Neurodyn. 2016;10(1):73–83.

51. Yao X, Liu Y, Lin G. *Evolutionary Programming Made Faster*. IEEE Trans Evol Comput. 1999;3(2):82–102.

52. *Optimization Modeling with LINGO* [Internet]. LINDO Systems Inc. 17th Edition.; 2019. Available from: https://www.lindo.com/