

Optimal operation of Mula reservoir with combined use of dynamic programming and genetic algorithm

Deepti Rani¹ · D. K. Srivastava¹

Received: 6 February 2015 / Accepted: 9 November 2015 / Published online: 20 November 2015
© Springer International Publishing Switzerland 2015

Abstract Genetic algorithm (GA) has been used repeatedly in reservoir operation studies during last two decades. GAs require trying different alternatives, different GA parameter values, and select those which perform best for a particular application. Besides this, there are chances of getting trapped into local optima, since GA starts with randomly generated initial population within the entire search space. Therefore, GA's search process is slow and time-consuming. GA's process may be speeded up if initial population is generated in a narrowed search space. This process may save time spent on sensitivity analysis of parameters. Discrete dynamic programming (DP) provides global optimal solutions, but discreteness and dimensionality are the main disadvantages in reservoir operation applications. To overcome these deficiencies a hybrid approach combining DP and GA (DP–GA) is proposed to study a single reservoir operation problem. Where DP provides the narrowed search space within which the optimal solution for the problem is expected. GA then search to achieve the possible optimal solution within this space avoiding any discreteness of variables. Proposed DP–GA approach was found to outperform both GA and DP in terms of less computational requirement and quality of the solution, respectively.

Keywords Genetic algorithm · Dynamic programming · Reservoir operation · Large size reservoir

Introduction

Over the last decade, evolutionary and meta-heuristic algorithms have been extensively used as search and optimization tools in various problem domains. Their applicability to wide-range of problems, easy to use, and global perspective are the main reasons which makes them preferred tools for extensive and successful applications to a variety of problems. GA belongs to the evolutionary algorithms (EA) family which has been used in many single and multireservoir operation studies (Jothiprakash and Shanthi 2006; Chen and Chang 2007; Yun et al. 2010; Hinçal et al. 2011; Scola et al. 2011, 2014). The evolutionary algorithms like GA, ant colony optimization (ACO) and particle swarm optimization (PSO) have great potential for handling non-linearity and solving multi-objective problems. Genetic algorithm has been applied to many real life optimization problems by several researchers. GA is search and optimization procedure that is motivated by the principle of natural genetics and natural selection (Davis 1991; Deb 2003; Goldberg 1989; Haupt and Haupt 2004; Holland 1975; Michalewicz 1996; Mitchell 1999).

Traditional GA or simple GA along with many attractive features poses computational difficulties, e.g., convergence speed to attain the optimum and getting stuck at local optima. Therefore, researchers have made efforts to modify GA itself or used it with other techniques to overcome these drawbacks. The main drawbacks are slow convergence, sensitivity analysis to find proper GA parameters so as to reach the optimum and getting struck into local optima, which is usually due to the initial population chosen or the parameters chosen. Ganji et al. (2007) developed a modified version of the simple GA, called SGA, for application to a reservoir operation problem. The SGA reduces the overall run time as compared to the

✉ Deepti Rani
deepinatyan@yahoo.com

¹ Department of Hydrology, IIT Roorkee, Roorkee, India

simple GA through dynamically updating the length of chromosomes. Karamouz et al. (2007) solved a similar problem using a GA-K nearest neighborhood (KNN) based optimization model. In this methodology, the lengths of chromosomes are increased based on the results of a KNN forecasting model. A hyper cubic distributed GA (HDGA) was applied by Chen and Chang (2007) with three general GA operators, linear ranking selection, blend- α crossover and Gaussian mutation, to search for the optimal reservoir release.

Hinçal et al. (2011) applied GA to a multireservoir system operation to maximize the energy production in the system by using two different approaches, conventional (monthly) and real-time approach. The comparison of the results revealed that the energy amounts optimized by using conventional approach were higher than the energy produced in real-time operation. However, by using real-time approach, a close approximation to the real operational data had been achieved.

The immune genetic algorithm (IGA), which is based on the immune network theory and GA, was introduced in the reservoir optimization field by Dong et al. (2010) as an adaptive immune GA (AIGA) for finding optimal reservoir dispatching. The main features of AIGA are the automatic changing of the crossover probability and mutation probability with respect to the fitness value of the objective function. A noisy GA (NGA) was proposed by Yun et al. (2010) to deal stochastic inflow pattern. They compared the efficiency of NGA with Monte Carlo method and concluded that the NGA shows better performance than the Monte Carlo optimization technique.

Generation of random initial population plays an important role in faster convergence and to reach global optimal solution for GA. Simple GA often suffers when initial random population is not so good. To increase the convergence speed of GA and to overcome getting stuck at local optima, Cheng et al. 2008 used chaos optimization algorithm (COA) for generating initial population for GA. They proposed integrated COA and GA approach, named chaos GA (CGA) and applied it for optimization of a hydropower generating reservoir system. They found CGA more feasible and effective in searching optimum as compared to the traditional GA.

It is reported in a recent review paper that combined or hybrid of GAs techniques shows better results than simply GA (Hossain and El-shafie 2013). Rani and Moreira (2010) reviewed both traditional and computational intelligence techniques used in reservoir operation studies they also ended up giving emphasis to investigate combined use of traditional and computational intelligence approaches to explore the unique advantages of these techniques. The main motive behind the hybridization of different techniques is to utilize their complementary characteristics, that

is, hybrids are supposed to give better results due to interaction of their respective features. In fact, choosing a suitable combination of complementary algorithmic concepts can be the key for achieving top performance in solving many difficult optimization problems. Owing to the drawbacks of traditional GA, researchers have made efforts to improve the technique itself by using the hybrid approach.

Cai et al. (2001) presented a combined genetic algorithm–linear programming (GA–LP) strategy for solving a large nonlinear reservoir systems optimization model. The hybrid approach was intended to overcome the limitation of LP and to improve the performance of the traditional GA. Use of GA was to linearize the original problem in each time period which was later solved sequentially using LP. The hybrid GA–LP approach was capable to find good approximate solutions to the nonlinear models. In view of computational advantages of combined GA–LP strategies to deal with nonlinearities, Reis et al. (2006) proposed and evaluated a stochastic hybrid GA–LP approach to operation of reservoir systems, which admits a variety of future inflow variability through a treelike structure of synthetically generated inflows.

Huang et al. (2002) presented a GA-based stochastic dynamic programming model (SDP) model to cope with the dimensionality problem of a multiple-reservoir system. A combination of GA and discrete differential dynamic programming (DDDP) was proposed by Tospornsampan et al. (2005) for irrigation reservoir operation problem. The main advantage of the hybrid approach is to save the computational resource for optimizing parameters. And also, since the good solutions obtained from GA are used as initial policy for DDDP, therefore reduces the probability of DDDP to trap in local optima. To utilize the benefits under safe flood control conditions, Liu et al. (2011) derived an optimal refill rule for a multi-purpose reservoir system. A hybrid multi-objective genetic algorithm (HMGA) is used in their study to integrate a progressive optimality algorithm into NSGA-II for a multi-objective reservoir.

In another popular method, simulated annealing (SA), a stochastic search technique is combined with the GA by Chiu et al. (2007) for application in reservoir optimization systems. The global search was undertaken in their study by the GA and the local search by SA. In their study, the process begins with the GA while they initiate a set of random solutions and after that selection, mutation and crossover operation is carried out for a few generations. Subsequently, all of the latest solutions from the GA are handled by SA to achieve further solutions. In such manner, the algorithm used GA and SA methodology together one after another. Li and Wei (2008) developed and compared the IGA-SA (improved genetic algorithm–

simulated annealing) with GA, DP and SA. The IGA-SA can help to overcome premature convergence faster than the earlier GA-SA algorithm and can escape the local optima. A hybrid approach, incorporating Incremental DP (IDP) and GA was proposed by Li et al. (2012). IDP was used to provide a rough optimal solution which was used for generating the initial population for GA. They concluded that, the GA optimization efficiency can be considerably enhanced and the difficulty of premature convergence that is common in the practical applications of GA to reservoir operation can be partially overcome. This hybrid IDP-GA approach was applied for daily operation of a two hydropower reservoirs. Authors showed that hybrid IDP-GA approach outperformed both the univocal IDP and the classical GA.

Jerma et al. (2013) demonstrated usefulness of a method based on the combined use of genetic algorithms (GA) and network flow optimization (NFO). The GA used was PIKAIA, and was linked to the SIMGES simulation model, a part of the AQUATOOL decision support system (DSS). The optimization of various operating rules was analyzed with the objective of minimizing short-term and long-term water deficits. They showed that simple operating rules produced similar results to the more sophisticated ones.

Modifications in the GA technique mainly improved the efficiency of GA for the particular problems studied and it may not necessarily be applicable to other problems therefore there is no unique method which may be applicable for all problems. So, research on hybrid GAs continues and further research scope still exists, especially for improvement in GAs or other EAs with the intension of increasing the convergence speed by finding better ways for generating initial population, reducing sensitivity analysis of operators and reducing premature convergence.

In this study an attempt is made to combine DP and GA to study a single reservoir operation problem. The methodology by combining DP and GA together is used to overcome some computational difficulties which are faced while applying DP or GA alone. GA is robust but suffers slow convergence and need extensive sensitivity analysis to optimize different parameters to reach optimum. DP has its own limits namely, discreteness and dimensionality, however can provide global optimal solution for discrete case, but the solution is optimal for the given increment and smaller increment size will give more precise solution but will increase computational difficulties. In combined approach, DP can provide the information of the potential regions of the search space as DP provides global optimal solution for discrete case. The initial population generated with this extra information about search space is an important help for the GA, and combined DP-GA

approach may lead to faster convergence and improvement in the quality of the solutions as compared to a simple GA.

Motivation for the proposed DP-GA approach

When a reservoir operation problem is solved using DP the storage (state variable) is divided in suitable discrete increments to solve the model, and optimization is performed conditionally on all combinations of storages. For a large size reservoir, the size of these increments will be bigger as to use very small increment size will pose computational difficulty. Although LP can handle a large number of variables but it cannot handle non-linearity, therefore LP is useful as long as the objective function and constraints are all linear. Addition of each increment increases the computational burden of DP multiplicatively. Although discrete DP will provide global optimal solution for given sets of states and increment size, the actual optimal may lie within the discrete interval between states (Fig. 1). This may not be a favoring result while the aim of the reservoir management is to efficiently and optimally use the available water, and the policy obtained based on these results may not be a good policy.

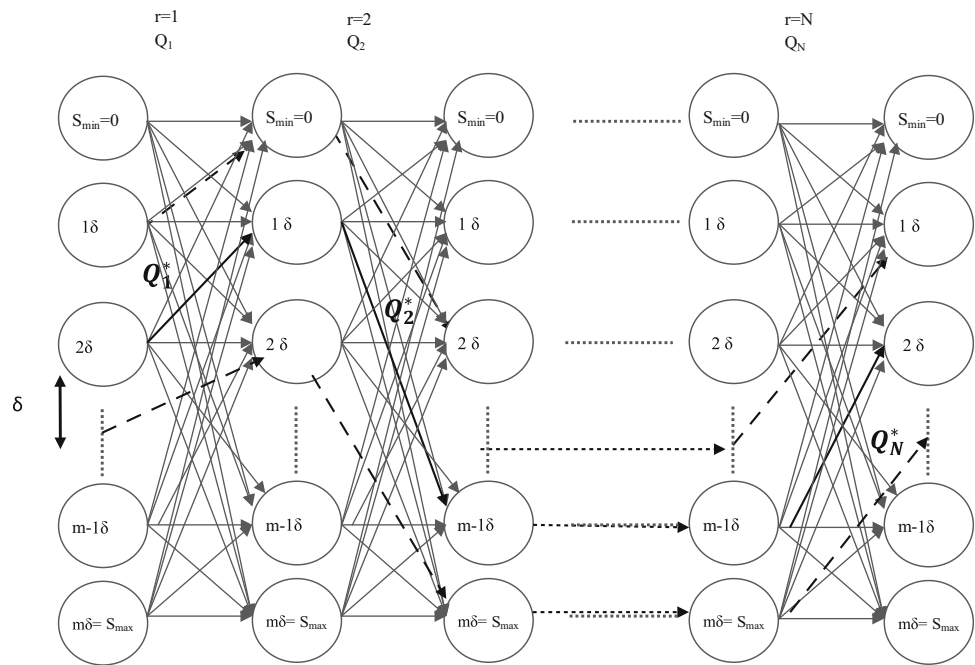
Heuristic optimization algorithms, such as, GAs are alternative tools for solving such large scale problems. However, for large size reservoirs the decision space becomes large because of a number of feasible solutions existing in the large search space. This may lead to premature convergence of these algorithms or computational difficulties, as it will require significant number of generations or large population size in addition to immense sensitivity analysis of GA parameters to achieve the optimal solution.

If initial search space for GA can be narrowed in a way, such that the optimal solution exists within this narrowed search space, computational requirement of GAs can be reduced up to large extent. The combined DP-GA is proposed to exploit the advantage of DP which can provide global optimal with less computational effort by using a coarse discrete increment size, thereby providing a potential region for the search space, to be further used by GA. This can further improve the solution with greater accuracy, avoiding any discreteness of decision or state variables. The DP-GA approach is used for a single reservoir operation by considering a large size reservoir of $608 \times 10^6 \text{ m}^3$ capacity, as case study.

Methodology

The DP-GA approach is devised in this paper for deriving optimal reservoir operation policy. The unique advantages of the DP and GA are combined to overcome their

Fig. 1 A reservoir operation problem fitted to conventional discrete DP framework



respective drawbacks. DP with suitable discrete interval size provides a global optimal solution to a problem. This global solution derived is for the discrete values of storages chosen. However, the actual optimal solution may fall within these discrete values of storages. Therefore, a search space could be defined around this solution. With the help of these results, a search space for initial population generation for GA search could be constructed. GA is then further used to find out a better optimum with continuous state and decision variables. This also avoids any discreteness in the variables. A flow chart of combined DP–GA procedure is presented in Fig. 2. After a DP-based search space is obtained, a certain number of unbiased initial individuals of GA are generated randomly within this space. The genetic operators including selection, crossover, and mutation are then used to optimize the objective function. By identifying the potential region for search, a large number of (1) non-optimal solutions and (2) needless computations of the GA, can be easily avoided. The prior selection of the required reasonable search space by the DP would (1) provide a narrowed corridor of search, (2) reduce the time required to reach to the optimal solution and (3) increase the overall computation efficiency.

Optimization with DP: finding the search space for GA

DP optimization is first applied to reservoir operation model to obtain an initial range for decision variables for the GA algorithm in this study. The DP model is applied to

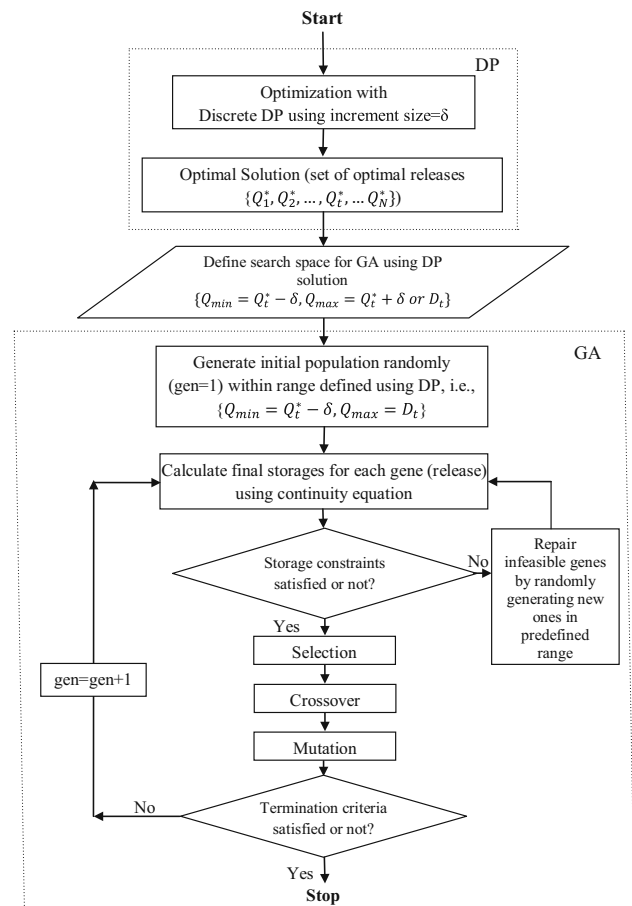


Fig. 2 Flow chart of combined DP–GA approach

estimate the optimal monthly releases, using monthly inflows to set a proper operation policy for the reservoir. The objective function is minimization of squared deviation of releases from target demands. In the DP model, the current inflow is the input to the reservoir; and the reservoir release includes the spill. For backward process in the DP, $f_r(S_r)$ is defined as the minimum optimal return function from operation of reservoir from all the r stages to go given that reservoir storage (state) in r stages to go is S_r ; $g_r(S_r)$ is the return function at r stages to go; S_r, S_{r-1}, I_r and EV_r are the initial reservoir storage (state), the resulting reservoir storage (state), the inflow to reservoir and the evaporation losses from reservoir in r stages to go, respectively; Q_r is the total reservoir release including spill, and is the decision variable for the DP model in r stages to go, respectively; D_r and Sp_r are the release target and the reservoir spill in r stages to go, respectively. Then the recursive equation for backward process for the DP models is:

$$f_r(S_r) = \min_{Q_r} [g_r(S_r, Q_r) + f_{r-1}(S_{r-1})] \quad \forall r = 1, \dots, N \quad (1)$$

The return function is:

$$g_r(S_r, Q_r) = (Q_r - D_r)^2 \quad (2)$$

Subject to the following constraints:

1. Reservoir continuity (state transformation) equation

$$S_{r-1} = S_r + I_r - Q_r - EV_r, \quad \forall r = 1, \dots, N \quad (3)$$

2. Spill from reservoir

$$Sp_r = Q_r - D_r, \quad \forall r = 1, \dots, N \quad (4)$$

3. Other constraints related to the upper bounds are on reservoir storages (states) limited to live storage capacity (Y_a) and non-negativity of state and decision variables.

$$Q_r \geq 0 \quad \forall r = 1, \dots, N \quad (5)$$

$$0 \leq S_{r-1} \leq Y_a \quad \forall r = 1, \dots, N \quad (6)$$

The whole reservoir operation process can be represented in DP framework as shown in the Fig. 1. Starting from known initial storage at $r = 1$, possible release decisions $\{Q_1, Q_2, \dots, Q_N\}$ at each stage should be such that in each stage the final storage belongs to the set of m predefined discrete states $\{0 \leq S_{\min} \leq m\delta\}$, and $S_{\max} = -Y_a$. Small value of ‘ δ ’ will increase the number of states if the reservoir is a large size reservoir, as a result computations of DP will increase. Conversely, a coarse increment size will reduce the accuracy of the optimal results, although the solution will be global optimal for a given increment. DP model solution provides a set of optimal releases $\{Q_1^*, Q_2^*, \dots, Q_r^*, \dots, Q_N^*\}$ which is the optimal policy for given discrete increment ‘ δ ’.

Optimization with GA

Real coded GA is used since floating point representation has advantages over binary, as it can represent the search space more precisely and closely to the problem (Michalewicz 1996). The objective function of GA is the same as in DP. The constraints are also same except the limits on releases, which are defined based on the search space obtained from DP model solution in case of DP-GA, while for simple GA the releases are limited by target demands.

The objective function is:

Minimization of squared deviation from target demands, i.e.,

$$g_t(Q_t) = \text{Min} \sum_{t=1}^N (Q_t - D_t)^2 \quad (7)$$

where $g_t(Q_t)$ is function of release at time period t . Q_t is the release for period t and D_t is target demand for time period t .

The constraints for the GA model are:

The continuity equation, which is stated as:

$$S_{t+1} = S_t + I_t - Q_t - EV_t \quad \forall t = 1, \dots, N \quad (8)$$

where S_t, I_t and Q_t are the storage, inflow and releases for the given reservoir at time period t and N is the time horizon for the problem under consideration. EV_t is evaporation from reservoir surface during time period t , respectively.

Limits on storage impose constraints of the form

$$S_{\min} \leq S_t \leq S_{\max} \quad \forall t = 1, \dots, N \quad (9)$$

Which ensure that the storage (S_t) will be within specified minimum and maximum storages.

Limits on release

$$Q_{\min} \leq Q_t \leq Q_{\max} \quad \forall t = 1, \dots, N \quad (10)$$

Release (Q_t) should be within specified minimum and maximum range.

Releases are the decision variables in this problem. Decision variables exist in the composition of the chromosomes of the population, $[\{Q_1^1, Q_2^1, \dots, Q_t^1, \dots, Q_N^1\}, \{Q_1^2, Q_2^2, \dots, Q_t^2, \dots, Q_N^2\}, \dots, \{Q_1^{NP}, Q_2^{NP}, \dots, Q_t^{NP}, \dots, Q_N^{NP}\}]$. Since initial population is generated within the limits specified for the decision variables, so the release constraints are readily satisfied (Eq. 10). In case of simple GA decision variables are generated within the entire range $\{Q_{\min} = 0, Q_{\max} = D_t\}$. The search space for DP-GA is defined as $\{Q_{\min} = Q_t^* - \delta, Q_{\max} = Q_t^* + \delta\}$, that is in δ vicinity of the optimal release made by DP (Fig. 1). To further refine this range the releases were generated within the range $\{Q_{\min} = Q_t^* - \delta, Q_{\max} = D_t\}$, i.e., the upper bounds on releases are kept as the target demands. Continuity

equation is also satisfied beforehand, since the storages are computed using the continuity equation itself given in Eq. (8).

Limits on storages are the only constraints which need to be examined for their feasibility during GA computations. Most popular approach to handle constraints in GA is using penalty functions for the violating constraints and then adding them to the objective function and then solving it as unconstrained problem. Many authors (Chang et al. 2010; Hınçal et al. 2011) have used penalty function approach for handling constraints in reservoir operation studies. The main problem in using this method is finding the appropriate penalty function associated to each constraint, which requires a sensitivity analysis or prior information about the constraints to set the penalty functions. Other approaches are replacing or repairing the chromosomes which create infeasibility or do not satisfy the constraints. The latter approach was found to outperform penalized approach by researchers (Alghazi et al. 2013; Michalewicz 1996). Repairing of chromosomes is adopted herein for dealing with infeasible chromosomes.

For handling storage constraints, following procedure is adopted. Assuming given initial storage at the beginning of the operation period for each chromosome, final storages are computed for each gene (release) using the continuity equation. Since, releases are generated randomly within specified limits; the corresponding storages may or may not lie within the range specified for them. A chromosome having genes (releases) which do not satisfy storage constraints is considered as infeasible chromosome and it goes through repairing process. During GA computations, if a storage value is less than the predefined lower limit, then the corresponding gene is repaired and a new gene is randomly generated in the search space, and is checked for its feasibility, if the corresponding storage found infeasible again then this cycle is repeated until a new gene is found which satisfy the lower limit on the storage. In case, the storage value exceeds the upper limit, the final storage will be equal the storage capacity of the reservoir and rest will be defined as spill (Sp_i) from the reservoir. The same procedure is repeated for all genes of each chromosome in the population to calculate final storages associated with each release until a feasible initial population is found. This population goes further through selection, crossover and mutation process to find new generation of the parent population. The new population is now checked for feasibility and refreshed again if needed before it goes through GA operators. This process is repeated for predefined maximum number of generations. The operators used in GA process are discussed below:

The main computational steps for the roulette wheel selection method are as follows:

1. Fitness of each chromosome, ft_i , and their sum $\sum_{i=1}^{NP} ft_i$ are calculated, where the population size is NP.
2. A real random number, $rand()$, within the range [0, 1] is generated for each chromosome and s is set to be equal to the multiplication of this random number by the sum of the fitness values, i.e., $s = rand() \times \sum_{i=1}^{NP} ft_i$.
3. The k^{th} chromosome is selected for the next generation, determining the value of k such that s is less than or equal to the cumulative sum of fitnesses up to k^{th} chromosome, that is, $s \leq \sum_{i=1}^k ft_i$.
4. Steps 2 and 3 are repeated until the number of selected chromosomes becomes equal to the population size, NP.

Crossover

Crossover operator can significantly influence the performance of real coded genetic algorithms as the exploration and exploitation of the search space is governed by crossover operator being applied. Generally, BLX- α crossover gives good results (Michalewicz 1996; Hınçal et al. 2011).

BLX- α crossover

Two offspring $y_1 = (y_1^1, y_2^1, \dots, y_n^1)$ and $y_2 = (y_1^2, y_2^2, \dots, y_n^2)$ are generated.

where, y_i^k is a randomly uniformly chosen number from the interval $[X_{\min} - I\alpha, X_{\max} + I\alpha]$

and X_{\max} , X_{\min} and I are defined as below

$$X_{\max} = \max\{x_i^1, x_i^2\}, \quad (11)$$

$$X_{\min} = \min\{x_i^1, x_i^2\}, \text{ and} \quad (12)$$

$$I = X_{\max} - X_{\min} \quad (13)$$

Higher value of α results in a better solution. As α is increased, the exploration level increases, since the relaxed exploitation zones spread over exploration zones, thereby, increasing the diversity levels in the population (Herrera et al. 1998).

Mutation

Random mutation is performed using a mutation probability which was fixed during sensitivity analysis of the parameters. In random mutation, the mutated gene is replaced with a new randomly generated gene within the specified search space.

Study area

The Mula project (Awchi 2004) envisages a major multi-purpose reservoir across the river Mula, a sub tributary of

the river Pravara in the Godavari river basin in the State of Maharashtra in India. It was initially constructed to serve the irrigation water needs of the system only. Subsequently, a provision for water supplies to Ahmednagar city, industries and villages were made. The total annual water requirements are $748 \times 10^6 \text{ m}^3$ out of which the annual irrigation water demand is $674 \times 10^6 \text{ m}^3$ and annual water supply requirements are $74 \times 10^6 \text{ m}^3$. A data series of 18-years-monthly inflow to the reservoir is available. The mean annual and the 75 % water year dependable inflows to the reservoir are 825×10^6 and $688 \times 10^6 \text{ m}^3$, respectively. Based on the historical data set, the estimates of the mean and the 75 % water year dependable inflows to the reservoir were obtained from the flow duration curve analysis. The gross reservoir capacity is $736 \times 10^6 \text{ m}^3$, active and dead storage capacities are 608×10^6 and $127 \times 10^6 \text{ m}^3$, respectively. There is no provision for flood storage at the project.

Results and discussion

The operation of the single reservoir Mula was taken as the case study to evaluate the performance of the DP–GA approach. The operation was done for an average year using the 18 years historical data. The average monthly flow and average monthly target demands are shown in Fig. 3. The DP–GA optimization results were compared with the DP and the simple GA. For DP model solution a computer code developed in FORTRAN was used. The DP model was solved to find an optimal policy with discrete state increment equal to $8 \times 10^6 \text{ m}^3$.

(1) The release policy using DP for an average year is shown in Fig. 4. Variations in the releases from target demands can be seen from this figure. The value of objective function is 70.31, which is the minimum squared deviation from target demands. For an average flow, there

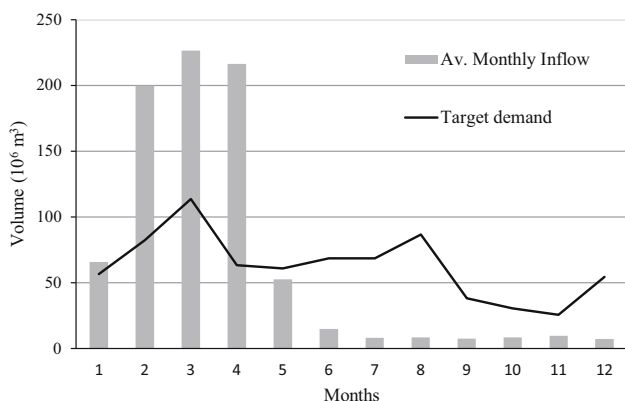


Fig. 3 Average monthly inflow and demand at Mula reservoir

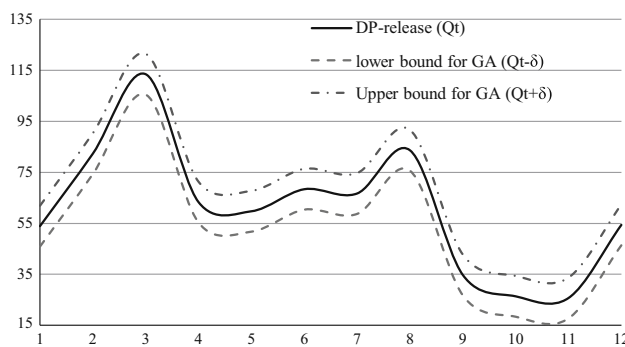


Fig. 4 Search space facilitated by DP for DP–GA

should not be any deficit as sufficient water is available for release, and this variation in releases is mainly due to the increment used in the DP. The DP underestimates or overestimates the decisions because of increment size used to discrete the storages or state variables. For large size reservoirs increment size is usually taken larger so as to contend computational difficulty of DP and the variation in releases may be more in such cases. Therefore, the results obtained are optimal for given increment but are not actually optimal as these results are further refined through simulation in practice. This can be done by using another approach which can search for the actual optimal within this space already searched by DP. In this study such an attempt is made by using the GA to refine the results obtained by the DP. The GA searches for the optimal in the space bounded by upper limit as $Q_t^* + \delta$ and lower limits as $Q_t^* - \delta$. The Fig. 4 shows the search space facilitated by the DP for the DP–GA approach.

This space can be decreased or increased depending on the problem. In present study, the search space was further refined as $\{Q_t^* - \delta, D_t\}$, as release should not be bigger than the target demand.

(2) GA and DP–GA approaches were also implemented through FORTRAN programs for solving the reservoir operation model. The code was first developed for simple GA and tested for some standard problems. It was further modified for reservoir operation problem to be solved using the simple GA and the DP–GA.

Sensitivity analysis of the GA parameters, e.g., population size, probability of crossover and mutation, etc., for fixing their best values, has been a common practice used by researchers in this kind of studies, e.g., Cheng et al. 2008; Chang et al. 2010; Li et al. 2012; Hidalgo et al. 2015 etc. Some recent studies have shown use of fine tuning of GA parameters in reservoir operation studies (Lerma et al. 2013; Scola et al. 2014), by incorporating routines for mutation and crossover probability optimization in GA code. Fine tuning of parameters may enhance the performance of GAs in general, but it requires long computer

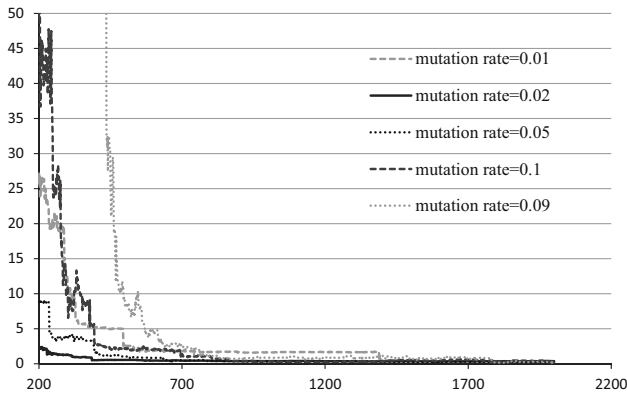


Fig. 5 Convergence with different crossover rates for population size = 50

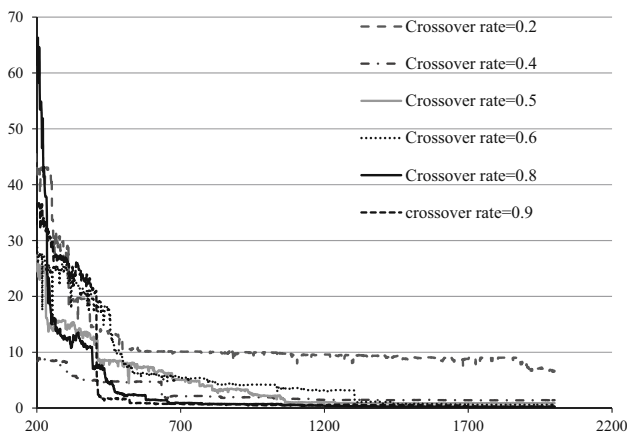


Fig. 6 Convergence with Mutation rate variation for population size = 50

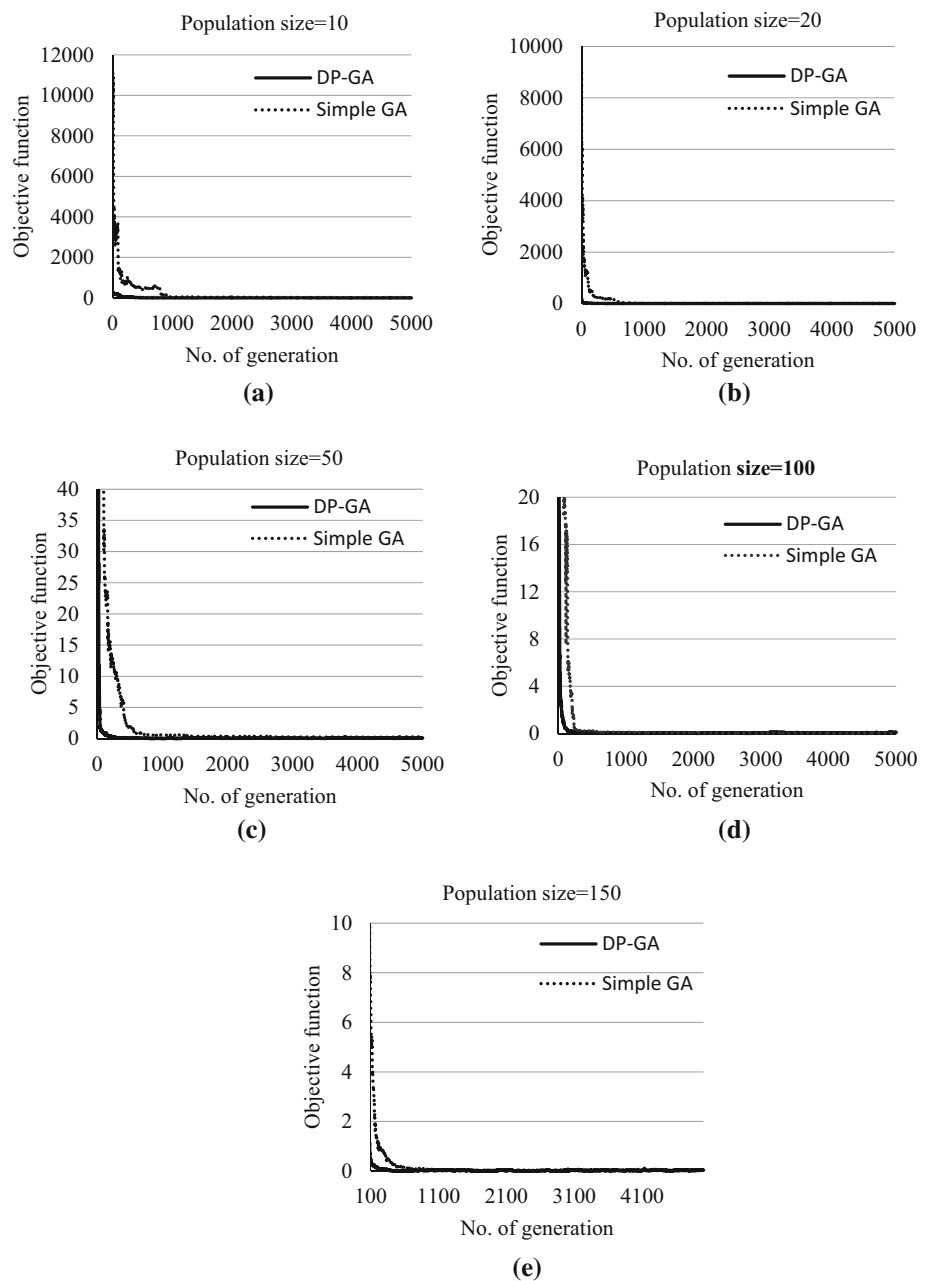
runs. In this study sensitivity analysis was followed and since the main aim was to examine the effectiveness of the hybrid approach and results were found worthy with the chosen values of crossover and mutation probabilities, fine tuning was not considered for optimizing GA parameters. For the BLX- α crossover operator the value of α was fixed at 0.5 after few trials. Crossover and mutation probabilities were also fixed after a few numbers of trials. For fixing the probability of crossover, a range 0.2, 0.4, 0.5, 0.6, 0.8, 0.9 of probabilities was tested and it was seen that better results were obtained for larger value of crossover probability, as suggested by many researchers too. Figure 5 shows the effect of crossover probability on performance of the GA for population size 50. Finally the probability of crossover was set to be 0.8. Different mutation probabilities were tried for population size 50, see Fig. 6. The mutation probability was found best as inverse of the population size. However, for present study it was taken as 0.01 in each case. In this study, however, no further sensitivity analysis was made on type of crossover or mutation operators and most commonly used selection, crossover and mutation methods were used to show the efficacy of the proposed hybrid approach.

(3) Population sizes and number of generations were varied to see their effect on optimization approach since large population size and large number of functions evaluations increases the computational requirement of the GA and make convergence slower. The limits on decision variables which was 0 to D_t for the simple GA and $\{Q_t^* - \delta, D_t\}$, for the DP-GA while all other parameters were kept same for the simple GA and the DP-GA to make a comparison.

Table 1 Performance evaluation of simple GA and DP-GA

Population size	Max. no. of generations	Generation no. when optimum reached		Value of objective function		Computation time (s)	
		GA	DP-GA	GA	DP-GA	GA	DP-GA
10	500	400	301	162.156	30.473	2–3	1–2
	1000	990	997	86.742	4.934	2–3	1–2
	5000	3753	1746	8.873	0.449	8–9	8
20	500	500	473	24.142	0.969	3–4	2–3
	1000	999	666	4.033	0.120	5	5
	5000	4217	2245	1.020	0.065	25	23–24
50	500	500	419	2.095	0.120	13	12
	1000	930	999	0.224	0.036	24	22–23
	5000	4208	1215	0.208	0.017	115	102
100	500	498	497	0.826	0.043	45	41–42
	1000	980	518	0.241	0.009	87	85
	5000	4179	2099	0.017	0.008	345	343
150	500	487	365	0.184	0.041	83	86
	1000	876	888	0.018	0.007	192	176
	5000	3155	2387	0.005	0.005	950	956

Fig. 7 a–e Convergence of simple GA and DP-GA with increase in population size



Results were obtained for population sizes 10, 20, 50, 100 and 150 using both the DP-GA and the simple GA. For each population size results were obtained for 500, 1000 and 5000 generations. Table 1 presents the value of objective function, number of generation where the optimum was found and computation time taken in each case. Convergence of the DP-GA and the simple GA is shown in Fig. 7a–e for different population sizes for 5000 functions evaluations. These figures clearly explain the difference in the performance of the two approaches. The DP-GA starts to converge towards optimum faster than the simple GA. Figure 8 shows that the optimal results for 100 population

size for 5000 generations using the simple GA are equivalent to the optimal results for population size 20 for the DP-GA for equal number of generations. The computation time for each case is 24 and 345 s, respectively. This shows that the DP-GA reaches to the same optimal more than 10 times faster. Likewise, it can be seen that similar results were obtained with population size 50 in 1000 generations or with a population size 100 in 500 generations (see Table 1; Fig. 8). In each case, it was observed that the DP-GA may possibly reach to optimal approximately 10 times faster as compared to the simple GA, either with small population size and larger number of generations or with

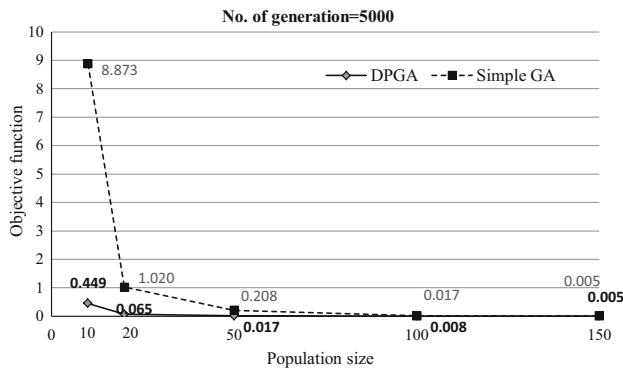


Fig. 8 Comparison of GA and DP-GA for different population sizes

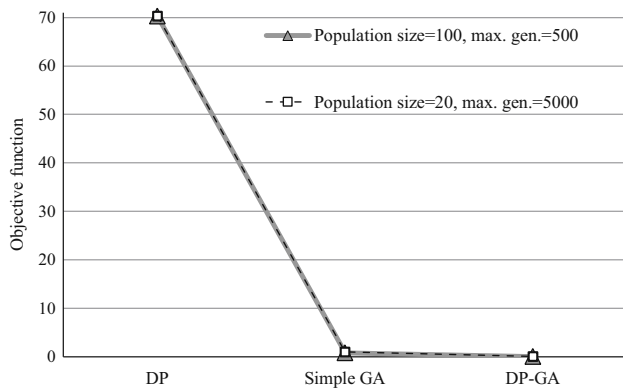
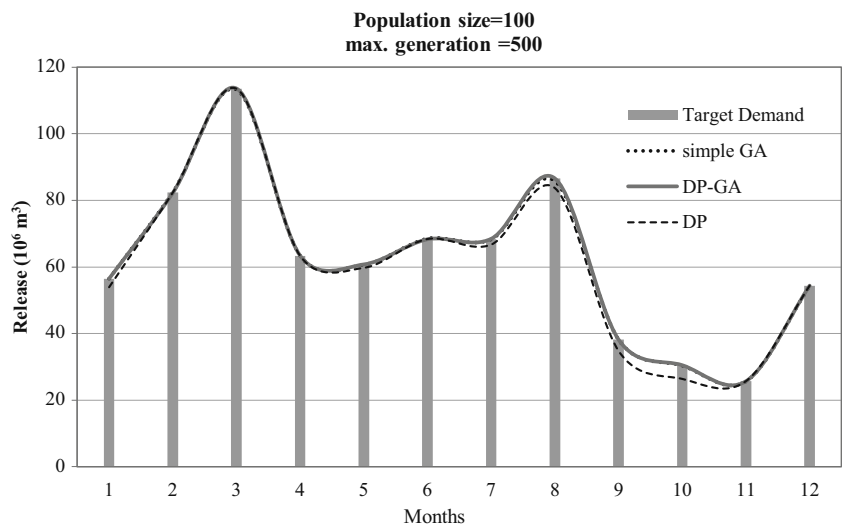


Fig. 9 Comparison of DP, simple GA and DP-GA performances

bigger population size and less number of generations (Fig. 9). In every case the computational requirement can be reduced approximately 10 times as compared to the simple GA. The DP-GA with less computational efforts gives better results as compared to the simple GA or the DP, as shown in Fig. 9. Similar results have been reported

Fig. 10 Release policy for DP, simple GA and DP-GA for case 1



in literature by Li et al. 2012, who used IDP and GA together for a cascaded hydropower system.

The optimal release policies obtained for the two cases from Fig. 9, that is, case 1. for population size 20, 5000 generations and case 2. population size 100, 500 generation using both the simple GA and the DP-GA are presented in Figs. 10 and 11, respectively. The monthly operation policies derived using both the DP-GA and the simple GA are very similar and better as compared to the DP. But, the computational requirement is considerably lessened using the DP-GA as compared to the GA. Since initial population is generated randomly, a number of solutions are obtained and the best solution is picked. Statistical performances of simple GA and DP-GA have been compared for ten different runs. The results are shown in Table 2 for the latter case, that is, population size 100 and 500 generations, for both simple GA and DP-GA. Results show that, the average values of the objective function obtained using GA and DP-GA are 0.951 and 0.056, respectively. The standard deviation of the solution in 10 different runs of the DP-GA algorithm is 0.011. The standard deviation of the objective function value calculated with GA was equal to 0.068, which is 6.4 times larger than that of DP-GA in 10 runs. Objective function values achieved using simple GA and DP-GA are of different scales and solutions with lesser value of objective function will give higher value of coefficient of variation, therefore, comparing their coefficient of variation is not worthy in this case.

Conclusions

The results of the combined DP-GA approach used to derive optimal reservoir operation policy of the Mula reservoir seems to be encouraging. The approach helped in

Fig. 11 Release policy for DP, simple GA and DP-GA for case 2

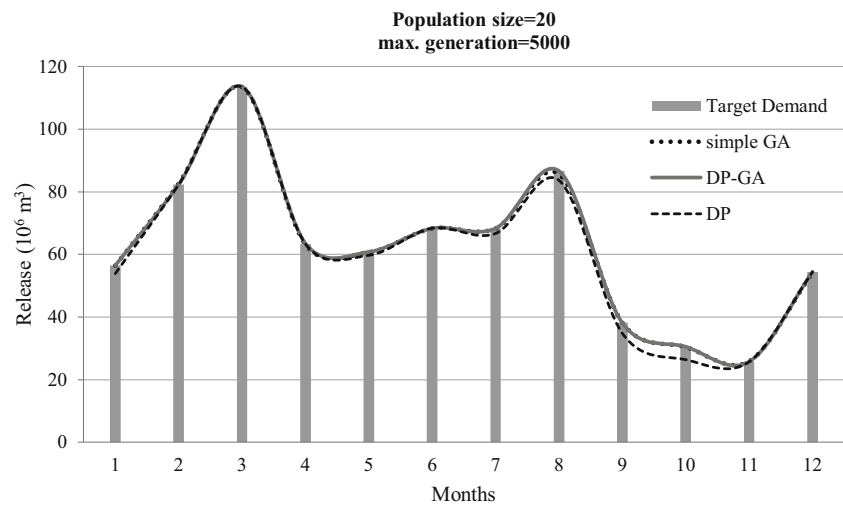


Table 2 Statistical comparison of simple GA and DP-GA performance

Number of run	DP-GA	Simple GA	DP
1	0.050	0.899	70.313
2	0.061	0.944	
3	0.083	0.866	
4	0.043	1.088	
5	0.058	0.959	
6	0.049	0.941	
7	0.051	1.039	
8	0.050	0.885	
9	0.060	0.938	
10	0.051	0.826	
Best	0.043	0.826	
Average	0.056	0.951	
Standard deviation	0.011	0.068	
Coefficient of variation	0.189	0.071	

reducing the computational requirements of the simple GA. The convergence of the DP-GA is approximately 10 times faster than the simple GA. Simple GA may achieve results equivalent to the DP-GA either by using a large population size and/or large number of generations. However, the convergence becomes quite slower in both the cases. In other words, the GA could achieve results comparable to DP-GA but at the cost of computational requirement. It can be said, therefore, that DP-GA outperforms the simple GA and DP in giving more precise results with less computational requirement. This approach is more useful for large size reservoirs where the decision space for the variables under consideration is large in size. In such cases a coarse increment size may be used to discrete state variables for the DP, while the GA may be used further for finding optimum within the narrowed space which has been

considerably reduced as compared to the entire search space. The proposed method shall be applicable to the problems which can be fitted into the DP framework. The operation policies derived using both the DP-GA and the simple GA are similar. But, the computational requirement is considerably lessened using the DP-GA as compared to the GA.

Acknowledgments Financial support provided by MHRD to work at IIT Roorkee is gratefully acknowledged by the first author.

References

- Alghazi A, Elazouni A, Selim S (2013) Improved genetic algorithm for finance-based scheduling. *J Comput Civil Eng* 27:379–394. doi:[10.1061/\(ASCE\)CP.1943-5487.0000227](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000227)
- Awchi TA (2004) Optimal planning and operation of a reservoir. Indian Institute of Technology, Roorkee
- Cai X, McKinney DC, Lasdon LS (2001) Solving nonlinear water management models using a combined genetic algorithm and linear programming approach. *Adv Water Resour* 24:667–676. doi:[10.1016/S0309-1708\(00\)00069-5](https://doi.org/10.1016/S0309-1708(00)00069-5)
- Chang L-C, Chang F-J, Wang K-W, Dai S-Y (2010) Constrained genetic algorithms for optimizing multi-use reservoir operation. *J Hydrol* 390:66–74
- Chen L, Chang FJ (2007) Applying a real-coded multi-population genetic algorithm to multi-reservoir operation. *Hydrol Process* 21:688–698. doi:[10.1002/hyp.6259](https://doi.org/10.1002/hyp.6259)
- Cheng C-T, Wang W-C, Xu D-M, Chau KW (2008) Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos. *Water Resour Manag* 22:895–909. doi:[10.1007/s11269-007-9200-1](https://doi.org/10.1007/s11269-007-9200-1)
- Chiu Y-C, Chang L-C, Chang F-J (2007) Using a hybrid genetic algorithm–simulated annealing algorithm for fuzzy programming of reservoir operation. *Hydrol Process* 21:3162–3172. doi:[10.1002/hyp.6539](https://doi.org/10.1002/hyp.6539)
- Davis L (1991) Handbook of genetic algorithms. Van Nostrand Reinhold, New York
- Deb K (2003) Multi-objective optimization using evolutionary algorithms. Wiley, Singapore

- Dong W, Zhang X, Liu X (2010) Adaptive immune genetic algorithm for optimization reservoir dispatching. In: International conference on computer and communication technologies in agriculture engineering (CCTAE), 12–13 June 2010. pp 41–44. doi:[10.1109/cctae.2010.5544227](https://doi.org/10.1109/cctae.2010.5544227)
- Ganji A, Karamouz M, Khalili D (2007) Development of stochastic dynamic Nash game model for reservoir operation II. The value of players' information availability and cooperative behaviors. *Adv Water Resour* 30:157–168. doi:[10.1016/j.advwatres.2006.03.008](https://doi.org/10.1016/j.advwatres.2006.03.008)
- Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading
- Haupt RL, Haupt SE (2004) Practical genetic algorithms. Wiley, Hoboken
- Herrera F, Lozano M, Verdegay JL (1998) Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. *Artif Intell Rev* 12:265–319. doi:[10.1023/a:1006504901164](https://doi.org/10.1023/a:1006504901164)
- Hidalgo IG, Correia PB, Arnold FJ, Estrócio JPF, de Barros RS, Fernandes JPT, Yeh WW-G (2015) Hybrid model for short-term scheduling of hydropower systems. *J Water Resour Plan Manag* 141(3):04014062
- Hinçal O, Altan-Sakarya AB, Metin Ger A (2011) Optimization of multireservoir systems by genetic algorithm. *Water Resour Manag* 25:1465–1487. doi:[10.1007/s11269-010-9755-0](https://doi.org/10.1007/s11269-010-9755-0)
- Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
- Hossain MS, El-shafie A (2013) Intelligent systems in optimizing reservoir operation policy: a review. *Water Resour Manag* 27:3387–3407. doi:[10.1007/s11269-013-0353-9](https://doi.org/10.1007/s11269-013-0353-9)
- Huang W-C, Yuan L-C, Lee C-M (2002) Linking genetic algorithms with stochastic dynamic programming to the long-term operation of a multireservoir system. *Water Resour Res* 38:1304. doi:[10.1029/2001wr001122](https://doi.org/10.1029/2001wr001122)
- Jothiprakash V, Shanthi G (2006) Single reservoir operating policies using genetic algorithm. *Water Resour Manag* 20:917–929
- Karamouz M, Mojahedi A, Ahmadi A (2007) Economic assessment of operational policies of inter-basin water transfer. *Iran Water Resour Res* 3:10–25
- Lerma N, Paredes-Arquiola J, Andreu J, Solera A (2013) Development of operating rules for a complex multi-reservoir system by coupling genetic algorithms and network optimization. *Hydrol Sci J* 58:797–812. doi:[10.1080/02626667.2013.779777](https://doi.org/10.1080/02626667.2013.779777)
- Li X-G, Wei X (2008) An improved genetic algorithm-simulated annealing hybrid algorithm for the optimization of multiple reservoirs. *Water Resour Manag* 22:1031–1049. doi:[10.1007/s11269-007-9209-5](https://doi.org/10.1007/s11269-007-9209-5)
- Li F-F, Wei J-H, Fu X-D, Wan X-Y (2012) An effective approach to long-term optimal operation of large-scale reservoir systems: case study of the three Gorges system. *Water Resour Manag* 26:4073–4090. doi:[10.1007/s11269-012-0131-0](https://doi.org/10.1007/s11269-012-0131-0)
- Liu X, Guo S, Liu P, Chen L, Li X (2011) Deriving optimal refill rules for multi-purpose reservoir operation. *Water Resour Manag* 25:431–448. doi:[10.1007/s11269-010-9707-8](https://doi.org/10.1007/s11269-010-9707-8)
- Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs, 3rd rev. and extended edn. Springer, New York
- Mitchell M (1999) An introduction to genetic algorithms. MIT Press, Cambridge
- Rani D, Moreira M (2010) Simulation-optimization modeling: a survey and potential application in reservoir systems operation. *Water Resour Manag* 24:1107–1138. doi:[10.1007/s11269-009-9488-0](https://doi.org/10.1007/s11269-009-9488-0)
- Reis LFR, Bessler FT, Walters GA, Savic D (2006) Water supply reservoir operation by combined genetic algorithm-linear programming (GA-LP) approach. *Water Resour Manag* 20:227–255. doi:[10.1007/s11269-006-8049-z](https://doi.org/10.1007/s11269-006-8049-z)
- Scola LA, Magela NO, Takahashi RHC, Cerqueira SAAG (2011) Multi-objective optimal multiple reservoir operation. In: Proceedings evolutionary computation (CEC), IEEE Congress, New Orleans, p 1927–1933
- Scola LA, Takahashi RHC, Cerqueira SAAG (2014) Multipurpose water reservoir management: an evolutionary multiobjective optimization approach. *Math Probl Eng* 2014:1–14
- Tospornsampan J, Kita I, Ishii M, Kitamura Y (2005) Optimization of a multiple reservoir system operation using a combination of genetic algorithm and discrete differential dynamic programming: a case study in Mae Klong system. *Thail Paddy Water Environ* 3:29–38. doi:[10.1007/s10333-005-0070-y](https://doi.org/10.1007/s10333-005-0070-y)
- Yun R, Singh V, Dong Z (2010) Long-term stochastic reservoir operation using a noisy genetic algorithm. *Water Resour Manag* 24:3159–3172. doi:[10.1007/s11269-010-9600-5](https://doi.org/10.1007/s11269-010-9600-5)