A New Dynamic Genetic Algorithm for optimising the reservoir operating rule curves: A case study of the Ubonrantana reservoir, Thailand

C. CHIAMSATHIT, A. J. ADELOYE & B. SOUNDHARAJAN

Institute for Infrastructure and Environment, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, UK

a.j.adeloye@hw.ac.uk

Abstract

A new dynamic genetic algorithm (DGA) was developed using a modified form of the search space reduction technique (SSRT). The new algorithm was applied to optimise rule curves for the Ubonratana multi-purpose reservoir in Thailand. Comparison of the new DGA and a standard GA showed that the new algorithm produced the optimal solution at half of the computation time required by the standard algorithm. Besides, the value of the optimum fitness function, i.e. the sum of the squared water deficits, was a mere 6,021 $(Mm^3)^2$, compared with 10,271 $(Mm^3)^2$ obtained with the standard algorithm. Finally, performance of the reservoir in terms of reliability, vulnerability, resilience and sustainability with the DGA derived rule curves was far superior to that of the standard algorithm.

Introduction

Reservoir operating rules are used for managing reservoir systems to achieve satisfactory performance in meeting the demands placed on them. To this end, various optimisation schemes have been used to derive optimal rule curves based on minimisation of water shortage related objective functions (Senthil Kumar et al., 2012). In particular, evolutionary genetic algorithms (GA) optimisation has long been recognized, and widely applied to provide the optimal solution when deriving reservoir operating policies (Chang et al., 2005). Standard GA (SGA), however, often fails to search adequately for the global optimum, especially when the search space is either too wide or too narrow. An excessively wide boundary will increase the computational time while a too narrow boundary may lead to the solution missing the global optimum (Purohit et al., 2013; Roeva et al., 2013). Thus, while a narrow boundary may be attractive in terms of computational time, due diligence is required to ensure that the boundary domain for the search does indeed contain the true optimal solution.

Consequently, researchers have applied search-space reduction techniques (SSRT) to improve and accelerate the search for the optimal solution from an initial wide boundary (Ndiritu and Daniell, 2001; Wu, 2002; Metkar and Kulkarni, 2013). For example, Liu (2012) used an adaptive boundary genetic algorithm to improve the precision of solutions (the best fitness function value) and speed up the convergence (the computational time) by moving the upper and lower boundaries of each generation around the mean value of the variable. However, if the random initial population is not in the optimal space, reducing the search space of the next generation, while reducing the search time might miss the global optimal solution by becoming trapped in a completely local optima. A new dynamic GA developed in this work overcomes this limitation of the traditional SSRT by ensuring that initial boundary is based on the current best fitness values, as will be discussed in the next section.

The aim of this work is to present a new development of the GA, known as the dynamic GA (DGA) that is more efficient than the standard GA (SGA) and represents an improved SSRT in arriving at an optimal solution. The objectives are to:

1. Review the literature dealing with the deployment of GA in reservoir optimisation;
2. Present the development of the new DGA optimisation and discuss its main features that distinguish it from the SGA;

3. Apply both SGA and DGA to the optimisation of rule curves for the operation of the Ubonratana multi-purpose reservoir in Thailand.

In the following sections, more details about reservoir rule curve optimisation using both SGA and DGA are given. The next section presents reservoir performance indices for assessing the effectiveness of the optimised rule curves. This is then followed by the case study after which the results are presented and discussed. The final section contains the conclusions.

Reservoir rule curves and its Genetic Algorithms optimisation

Rule curves

Rule curves are used to guide monthly decisions on water release from a reservoir. Fig. 1 illustrates how the rule curves are used to guide the reservoir operation. The flood control rule curve (FCRC) controls the level of discharge during floods. When the water level is higher than FCRC, the excess water must be discharged through the spillway to restore reservoir level to FCRC. The upper rule curve (URC) and lower rule curve (LRC) defines the maximum and minimum level for conservation purposes, respectively. The normal water level (NWL) defines the reservoir crest level of the spillway. The minimum storage (Min.WL.) defines the minimum water level for water supply or top of inactive zone. If the reservoir storage level is at or above URC, then the demanded water or possibly more must be released to restore the storage to URC. When the storage is between the URC and LRC, attempt is made to supply the full demand if possible; otherwise enough water that leaves the storage at LRC is released. If the reservoir level is at or below LRC, then no water will be released.

Fig. 1 schematically illustrates the above release patterns, where D_t and D'_t are the demand and actual release, respectively. The monthly ordinates of the URC and LRC were optimised using genetic algorithms.

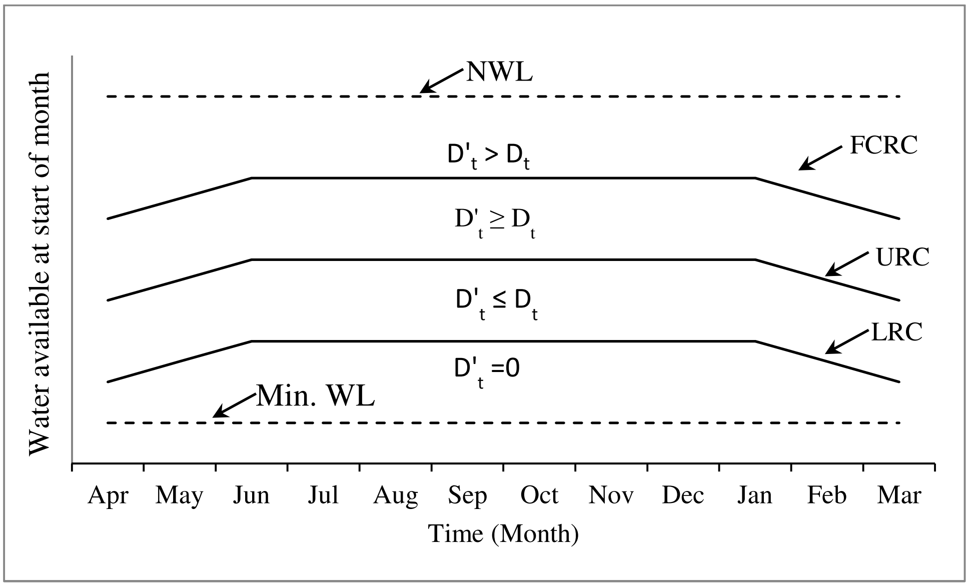


Figure 1 Schematic illustration of rule curves for reservoir operation

GA optimisation of rule curves

GA is an efficient, adaptive and robust population based optimisation method that uses the principles of natural selection and evolution. In this study, GA was used to optimise the ordinates of the upper (URC) and lower (LRC) rule curves for each month (see Fig. 1).

The objective function of the optimisation is to minimise the sum of squares of the period water shortages i.e. (Chiamsathit et al., 2014):

$$\text{Minimise} \quad \sum (D_t - D'_t)^2, \ \forall D'_t \le D_t \tag{1}$$

The continuity constraint or water balance constraint applied in this study (McMahon and Adeloye, 2005), which can be expressed as:

$$S_{t+1} = S_t + Q_t - D'_t - E_t \tag{2}$$

where S_t is storage at beginning of time t; S_{t+1} is the storage at the end of time t; Q_t is the inflow to the reservoir during t; E_t is the net evaporation (evaporation minus direct rainfall) in period t; and all other symbols are as defined previously.

The net evaporation loss E_t (volumetric unit) in any period t is taken to be the product of the average reservoir surface area during the period and the net evaporation rate (e_t) for that period, i.e.

$$E_t = e_t [(A_t + A_{t+1}) / 2] \tag{3}$$

$$e_t = EP_t - P_t \tag{4}$$

where, e_t is the net evaporation measured in equivalent depth of water; P_t is the rainfall during t; EP_t is the evaporation during t; A_t and A_{t+1} are the reservoir surface areas at the beginning and end of period t, respectively.

For planning purpose, a linear approximation to area-storage relationship is often assumed which can be expressed as:

$$A_t = aS_t + b \tag{5}$$

where, a and b are coefficients of the linear approximation to area-storage relationship. After substituting (5) and (3) in (2) and re-arranging, the mass balance equation becomes:

$$S_{t+1} = (S_t(1 - 0.5ae_t) + Q_t - D'_t - be_t)/(1 + 0.5ae_t) \tag{6}$$

As illustrated in Fig. 1, water release is based on the amount of water available at the start of the month relative to the ordinates of the rule curves. The amount of water available, WA_t is given by:

$$WA_t = S_t + Q_t \tag{7}$$

The three possible cases are:

Case 1: For $WA_t \ge URC_m$ this is the excess operation case.

$$D'_t \ge D_t \tag{8}$$

$$D'_t = S_t + Q_t - E_t - URC_m \tag{9}$$

$$Y_t = D'_t - D_t \tag{10}$$

Case 2: For $LRC_m < WA_t < URC_m$ this is the normal operation case.

$$D_t' \leq D_t \tag{11}$$

$$Y_t = 0 \tag{12}$$

$$\text{If} \quad WA_t - D_t \geq LRC_m, D_t' = D_t \tag{13}$$

$$\text{If} \quad WA_t - D_t < LRC_m, D_t' = WA_t - LRC_m \tag{14}$$

Case 3: For $WA_t \leq LRC_m$ this is the deficit operation case.

$$D_t' = 0 \text{ (No water released)} \tag{15}$$

where URC_m is the upper rule curve during month *m(=1, 2.,3,…,12)* of the year*;* LRC_m is the lower rule curve during month *m;* Y_t is the excess water released during period *t*. In general, t = 12(y-1) + m for years y = 1, 2, 3…., n, where n is the number of years in the data record.

Standard Genetic Algorithms (SGA)

GA is an efficient, adaptive and robust population-based optimisation method that uses the principles of natural selection and evolution. The SGA is implemented according to the schematic in Fig. 2. It starts with an initial population of the solutions, i.e. the ordinates of the URC and LRC in this study, which is generated randomly. The reservoir simulation then takes place following which the associated deficits are used to compute the objective function (Eq. (1)) and hence isolate the fittest solution in the population. Genetic operations- selection, crossover and mutation (Michalewicz, 1992)- are then carried out to create a new generation of solution population. This new generation undergoes similar "fittest" solution identification, and the whole process is repeated over several generations until the stopping criterion is met, at which point the optimum solution is said to have been reached. Because the GA is initialised with random numbers which are unlikely to be the same over repeated trials or sets, the algorithm in Fig. 2 is normally repeated several times, typically 100, and either an average solution or the best among the set of 100 taken. Factors that affect the convergence include the number of generations, the population size and the number of repetitions or sets.

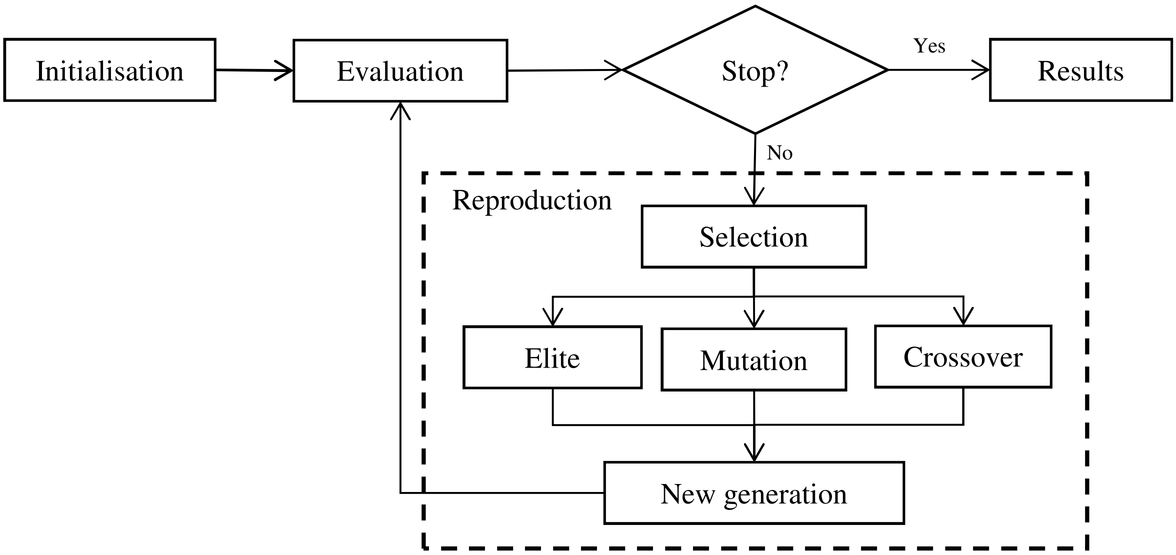


Figure 2 Standard genetic algorithm flowchart

One of the key GA parameters is population size (number of chromosomes). The population size specifies how many solutions are in each generation. With a large population size, the algorithm could search more points and thereby obtain a better result (Purohit et al., 2013). However, an excessively small population could guide the algorithm to poor solutions, while an excessively large population could significantly increase the computational time in finding a solution (Roeva et al., 2013). The SGA algorithm investigated the effect of population sizes (50, 100, 200 and 250) and generations (1-3000) on the fitness values. The values adopted for other key GA's parameters and settings in this study are shown in Table 1.

Table 1 The GA's parameters and settings

Key parameters	Value
Selection	roulette-wheel
Elite count	1
Crossover operator	Scattered crossover
Crossover fraction	0.8
Mutation	Uniform
Mutation rate	0.01

Dynamic genetic algorithm (DGA)

The DGA is schematically illustrated in Fig. 3 also based on its use for optimising reservoir rule curves. It starts with an initial random population like the SGA and runs over "g" generations from which the best string is selected. This process is repeated "r" times, thus leading to "r" best strings. "g" and "r" are parameters of DGA and their best values were determined by trial-and-error but, as will be seen later, are much lower than those normally required for the SGA. As noted for the SGA, the number of generations can be as high as 3000 and the number of sets (or repetitions) as high as 100. The best of "r" strings are then observed for the purpose of updating the boundaries for the search space for the next iteration. The best value for any two consecutive iterations are compared and it the difference is above a specified value "β" , new boundaries are specified and the process is repeated.

The DGA algorithm was tested for generations "g" (= 2, 5, 10, 15 and 20) and repetitions "r" (= 3, 5, 7 and 10). The stopping criterion β, i.e. the difference between the best fitness values from two consecutive sets (k^{th} and $K+1^{th}$), was set to 0.05. In the case of deriving the ordinates of the upper and lower rules for a reservoir illustrated in Fig. 3, the new boundaries are located between the maximum and minimum of the best strings value in the current set (k) plus/minus the average of the range of the string value in the previous set (k-1). Thus, mathematically, the boundary settings become:

(i) For k=1 is the 1^{st}set

$$UB_{i,1} = X\,max_{i,1}, LB_i \le UB_{i,1} \le UB_i \tag{16}$$

$$LB_{i,1} = X\min_{i,1}, LB_i \le LB_{i,1} \le UB_i \tag{17}$$

(ii) After the 1^{st} set or k>1

$$UB_{i,k} = Xb_{i,k} + 0.5(X\max_{i,k-1} - X\min_{i,k-1}), LB_i \le UB_{i,k} \le UB_i \tag{18}$$

$$LB_{i,k} = Xb_{i,k} - 0.5(X\max_{i,k-1} - X\min_{i,k-1}), LB_i \le LB_{i,k} \le UB_i \tag{19}$$

The algorithm is stopped at

$FVAL_{k-1} - FVAL_k \le \beta$, and the best performing string of the last set is the final result.

where UB_i, $UB_{i,1}$ and $UB_{i,k}$ are respectively the initial, 1st set and kth set upper boundary of ith variable (i=1 to 24 variables); LB_i, $LB_{i,1}$ and $LB_{i,k}$ are respectively the initial, 1st set and kth set lower boundary of ith variable; $Xb_{i,k}$ is the best value of the ith decision variable in the kth set; $Xmax_{i,1}$ and $Xmax_{i,k-1}$ are respectively the maximum of ith variable in the 1st set and k-1th set; $Xmin_{i,1}$ and $Xmin_{i,k-1}$ are respectively the minimum of ith variable in the 1st set and k-1th set; β is the difference between the values of the best fitness value in the k-1th set ($FVAL_{k-1}$) and the kth set ($FVAL_k$).

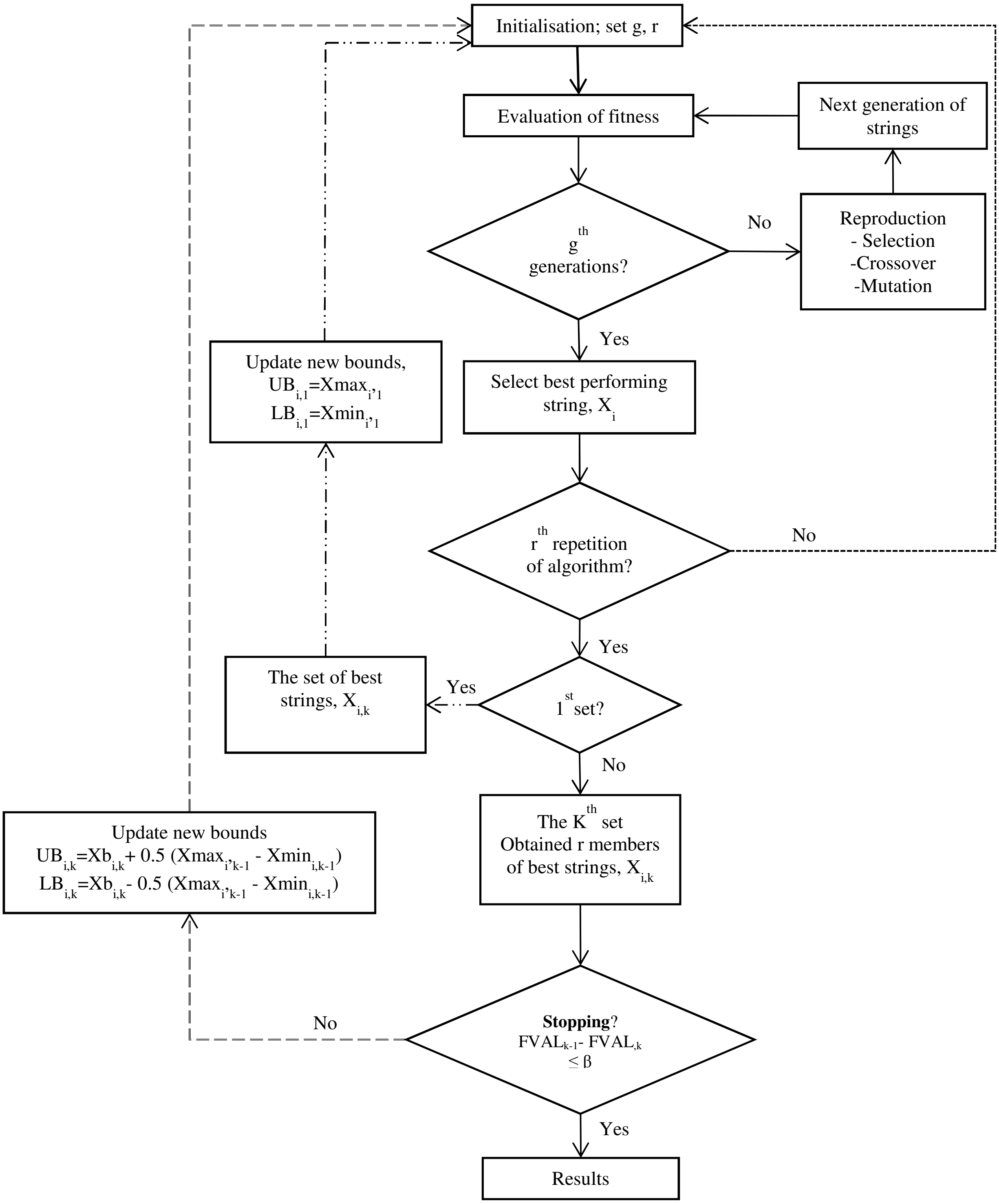

Initialisation; set g, r
Evaluation of fitness
Next generation of strings
No
Reproduction
- Selection
-Crossover
-Mutation
g^{th} generations?
Yes
Update new bounds, $UB_{i,1}=Xmax_{i',1}$ $LB_{i,1}=Xmin_{i',1}$
Select best performing string, X_i
r^{th} repetition of algorithm?
No
Yes
The set of best strings, $X_{i,k}$
Yes
1^{st} set?
No
Update new bounds
$UB_{i,k}=Xb_{i,k}+ 0.5 (Xmax_{i',k-1} - Xmin_{i,k-1})$
$LB_{i,k}=Xb_{i,k}- 0.5 (Xmax_{i',k-1} - Xmin_{i,k-1})$
The K^{th} set Obtained r members of best strings, $X_{i,k}$
No
Stopping?
$FVAL_{k-1}- FVAL_{,k} \leq ß$
Yes
Results

Figure 3 Dynamic genetic algorithm flowchart

Evaluated Performance Indices

To test the effectiveness of the optimised rule curves, monthly reservoir simulations were carried out and relevant performance measures- reliability (time- and volume-based) and vulnerability (McMahon *et al.*, 2006; Adeloye, 2012)- were evaluated as outlined below.

i. *Time-based Reliability* (R_t) is the proportion of the total time period under consideration during which a reservoir can able to meet the full demand without any shortages:

$$R_t = N_s / N \tag{20}$$

where N_s is the total number of months out of N that the demand was met.

ii. *Volume-based Reliability* (R_v) is the total quantity of water actually supplied divided by the total quantity of water demanded during the entire operational period:

$$R_v = \sum_{t=1}^{N} D_t^{'} \Big/ \sum_{t=1}^{N} D_t \,,\ \forall D_t^{'} \le D_t \tag{21}$$

iii. Resilience is a measure of the reservoir's ability to recover from failure and the most widely used definition of resilience is attributable to Hashimoto et al. (1982)

$$\varphi = \frac{1}{\left(\frac{f_d}{f_s}\right)} = \frac{f_s}{f_d};\ \ 0 < \varphi \le 1 \tag{22}$$

iv. *Vulnerability* is the average period shortfall as a ratio of the average period demand (Sandoval-Solis *et al.*, 2011):

$$\eta = \frac{\sum_{t=1}^{f_d} [(D_t - D_t^{'}) / D_t\ ]}{f_d} ; t \in f_d \ ,\ \forall D_t^{'} \le D_t \tag{23}$$

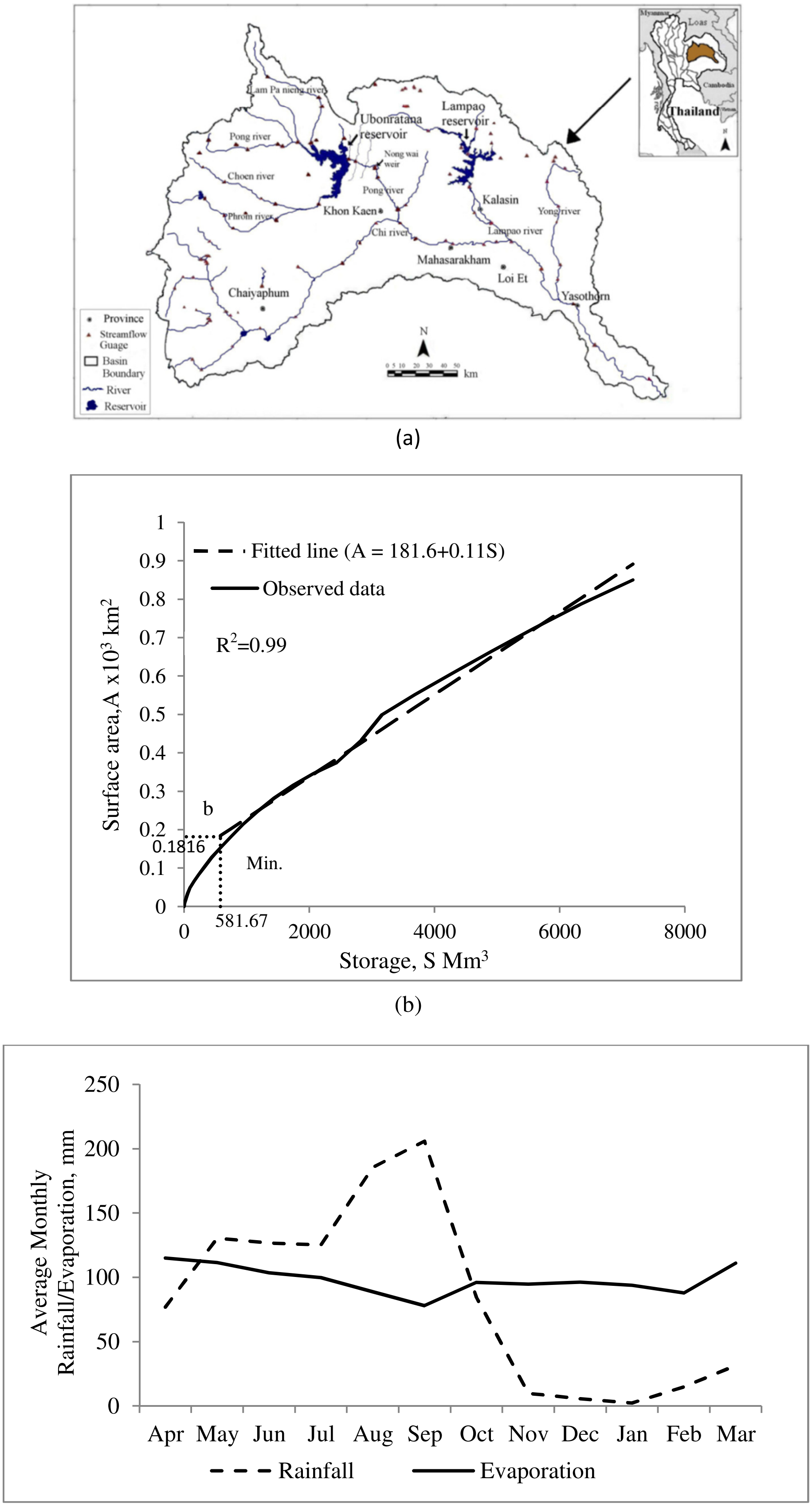
where η is vulnerability (dimensionless), f_s is the number of failure sequence, f_d is the total duration (months) of the failures, i.e. $f_d = N - N_s$ and all other terms are as defined previously.

v. *Sustainability index* integrates the three earlier defined indices was recently proposed by Sandoval-Solis et al. (2011):

$$\lambda = \left(R_t \varphi (1-\eta)\right)^{1/3} \tag{24}$$

Case Study and Data

The methodology was applied to the single, multi-purpose Ubonratana reservoir in the upper Chi River basin in north-eastern Thailand (Fig.4a). Purposes served by the reservoir are public water demand, downstream water requirement and irrigation. Gross water requirements for the 384 months were 30,140 Mm^3, i.e. annual average public demands of 12 Mm^3, annual average downstream requirements of 224 Mm^{3}and an annual average irrigation demand of 706 Mm^3. Where available water is insufficient, water allocation at Ubonratana is prioritised in the following order: public demands (domestic and industrial demand), downstream requirements (minimum in-stream and other agriculture requirement) and irrigation demand, respectively.



(a)

(b)

(c)

Figure 4: Study location showing: (a) map of Thailand, Chi River basin (Chiamsathit et al., 2014); (b) derived reservoir surface area-storage relationship for Ubonratana reservoir; (c) average monthly rainfall and evaporation distribution.

The dam is located on Pong River at Phong Neap, Ubonratana district in Khon Kaen province, between latitudes 16° and 17°30'N and longitudes 101°15" and 102°45" E. The reservoir was completed in 1966 and started operation in 1970 with its catchment area of 1200 km^2 for water supply (domestic, industrial, irrigation, and downstream needs), hydropower generation and flood control. The single, multi-purpose reservoir has been operated for a long time using rule curves developed by the Electricity Generating Authority of Thailand (EGAT), the dam operators. The reservoir has a storage capacity of 2,431Mm^3 and a hydropower generating capacity of 25.2 MW. The dam height is 36 m, with a length of 885 m (including the 100 m spillway) and a width of 6 m at the top (EGAT, 2002). The minimum storage volume control is 581 Mm^3 for generating hydropower and 120 Mm^3 for dead storage (EGAT, 2002). All the water deliveries first pass through the turbines for power generation before being allocated to the other uses. The area–storage relationship is shown in Fig.4b. As noted previously, Fig. 4b was used for incorporating evaporation loss in the simulation.

The study used the reservoir inflow data of 384 months (1980-2012) provided by EGAT, the dam's operators, and current demand data were provided by the Royal Irrigation Department (RID) of Nong Wai. The average annual inflow is 2619 Mm^3 includes the runoff and direct rainfall on the reservoir surface. The inflows were not measured directly but were estimated by EGAT using mass balance considerations (EGAT, personal communication). The average annual rainfall is 1200 mm and its monthly distribution is shown in Fig 4(c) which also contains the monthly mean evaporation rates. The rainfall and evaporation data of 21 (1988-2008) years were also provided by Royal Irrigation Department (RID) of Nong Wai, that have responsibility for water allocation. Because of the shortness of the rainfall and evaporation data relative to the reservoir inflow data, only the mean monthly values of both the evaporation and rainfall were used for the reservoir simulations. As previously found out by Fennessey (1995), using mean values of the net evaporation in reservoir simulations produced no significant difference from using time series data of net evaporation.

The climate in this region is normally divided into 3 seasons: summer (February-April), rainy (May-September) and winter (October-January). For cultivation purpose, the crop growing periods consist of wet period (June-October) and dry period (December-March).

Results and Discussion

Influence of the population size and the number of generations on GA's performance

Fig.5 shows the effect of population size on the fitness values for the SGA, from where it is clear that increasing the population above 200 does not produce any significant improvement in the fitness function. The algorithm has been run 30 times for each case to accommodate the variability associated with the random generation of the initial solution populations. The fitness plotted in Fig. 5 represents the mean for the 30 repetitions. A population of 200 was thus adopted to test the effect of the number of generations on the fitness and the result is shown in Table 2. As shown in Table 2, while the fitness function reduced by 14.9% when the generation was increased from 100 to 1500, increasing the generation beyond 1500 produced no noticeable improvement. Thus, for the SGA implementation in the Ubonratana rule curves optimisation, it would seem that a population size of 200 and 1500 generations are the best combination.

The complete set of the best fitness values for all 30 runs for this combination of population size and generation is shown in Fig. 6. This clearly demonstrates the variability in the best solution as expected, given the random nature of the initial solution population. The minimum best fitness in Fig. 6 was 10271 while the maximum was 29671. The computation time, although not plotted here, was equally variable with mean = 792 secs; max = 823 secs and min = 733 secs.

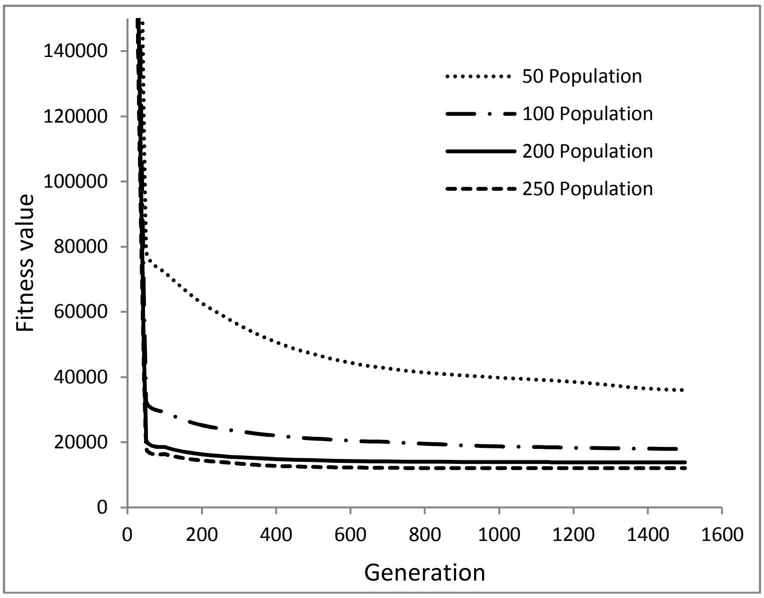


Figure 5: Comparison of the effect of population size

Table 2 Influence of number of generations on the fitness value for SGA (population = 200)

Generations	100	1500	3000
Fitness value	18569.12	15809	15761.48

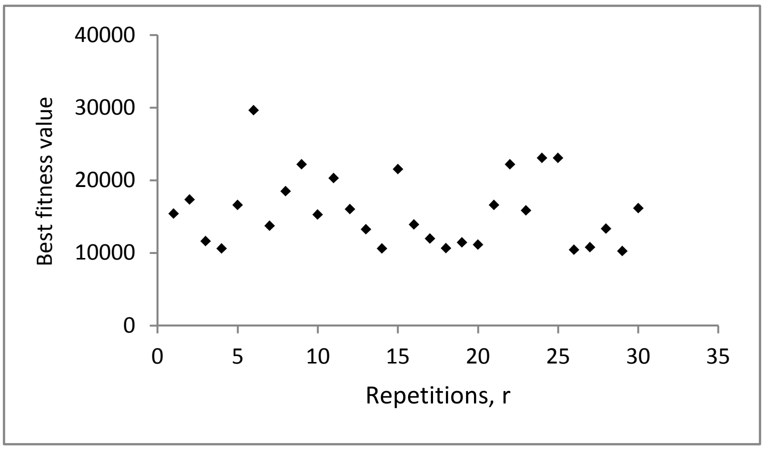


Figure 6: The fitness values of the algorithm of 30 repeating times in SGA

Influence of the generations and the repetitive algorithm on DGA's performance

The effect of the number of generations (g) and repetitions (r) on the computation time is shown in Table 3, while Fig. 7 depicts the variations in the fitness function as both the 'g' and 'r' change. As expected, increasing both the 'g' and 'r' causes the computation time to increase. However, much more significant for this work is the influence of 'g' and 'r' on the fitness function. As Fig. 7 shows, the global minimum of the fitness function was 6052 but required about 20 generations to attain with r=3 or 5, this global minimum was reached after only 2 generations for r=7. In fact, increasing the repetitions to r=10 produced a result that is indistinguishable from that of r=7. This implies that “g”=2 and “r”=7 represents the best combination in DGA. The best fitness value for the “r”=2 and “g”=7 combination in DGA as shown in Fig.7 was 6021, which is about 43% of that achieved with the SGA. Additionally, the computational time for the best DGA was 352 seconds (as seen in Table 3), i.e. less than 50% of time taken by the SGA.

Table 3 Computation time (sec) for different “r” and “g”

Generations, g	Repetition, r			
	3	5	7	10
2	115	197	352	456
5	194	304	423	601
10	279	481	601	739
15	396	620	645	966
20	343	657	461	620

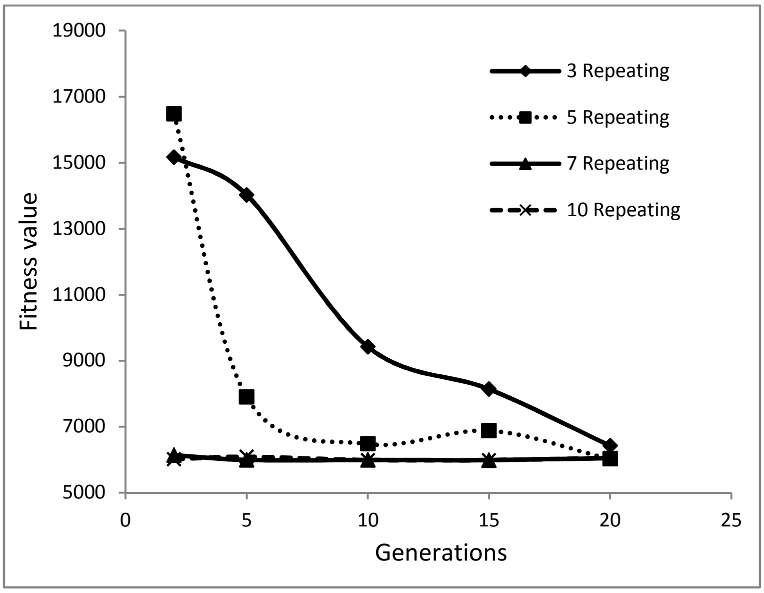


Figure 7: Influence of the generations and repetitive algorithm on the fitness value

Application of Standard GA and Dynamic GA for optimizing the operating rule curves

The ordinates of the optimised rule curves are listed in Table 4; the FCRC is the flood control rule curve which has not been optimised in this study but based on the values provided by EGAT. Figs 8a&b are the graphical illustration of the optimised rule curves using SGA and DGA, respectively. The optimal rule curves trajectories obtained from both SGA and DGA were well-behaved, with the nadir occurring around July/August so as to accommodate the large runoff during the Monsoon thus contributing to flood alleviation.

Table 5 summarises the reservoir performance (in terms of failure duration in months (f_d), volume based reliability, R_v (%); time-based reliability, R_t (%); resilience, φ; vulnerability, η; and sustainability, (λ) for each of the purposes of Ubonratana. Additionally, the sectoral sustainability indices were combined to obtain the group sustainability index for the entire water resources system using:

Table 4 Ordinates of rule curves (Mm^3) tested by SGA and DGA

GA	Rule Curve	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
SGA	FCRC	1662	1616	1571	1536	1527	1527	1902	1902	1852	1804	1756	1709
	URC	1141	1176	991	846	946	1340	1808	1901	1746	1366	1216	1556
	LRC	583	582	592	626	606	824	947	889	962	806	724	651
DGA	URC	1246	941	888	833	946	1423	1750	1750	1750	1422	1098	991
	LRC	583	582	582	603	582	625	961	911	848	783	714	651

$$\lambda_G = \sum_{j=1}^{M} w_j \lambda_j \tag{25}$$

where w_j is a weight. A simple way of specifying the weighting is to use the proportion of the total system average annual demand that is represented by each users category (Sandoval-Soils et al., 2011), i.e.

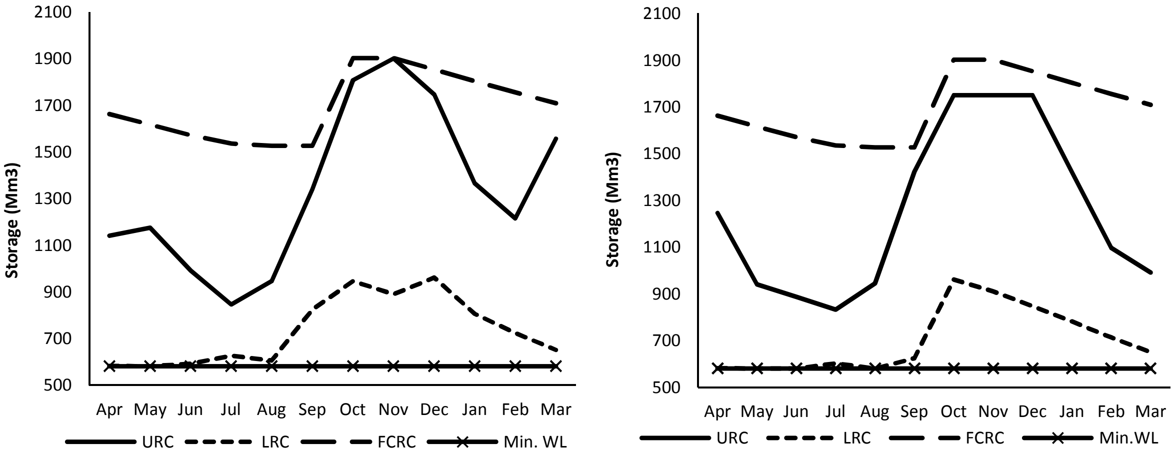
$$w_j = \frac{D^j}{\sum_{j=1}^{M} D^j} \tag{26}$$

where λ_G is the group sustainability; λ_j is the sustainability for users category j; w_j is the weighting for user j; M is the total number of users sectors and D_j is the average annual water demand for users sector j.

As seen in the Table 5, all performance indices of the reservoir using DGA were better than those of the SGA. The total unmet demand in SGA and DGA were 251.4 and 226.7 Mm^3, respectively. However, the time-based and volumetric reliability in DGA are marginally better than SGA. The vulnerability of the optimised policy using DGA is much better than using SGA in public and downstream supply sectors. The optimised policy using DGA offers a system that is almost 46.5% more sustainable than the optimised policy using SGA.

Table 5 Summary of evaluated reservoir performance indices for the optimised policies

Indices/	SGA			DGA		
Water user	Public	Downstream	Irrigation	Public	Downstream	Irrigation
Total period delivery (Mm^3)	356.8	7165.6	22366.2	357.2	7179.8	22376.3
Total period deficit (Mm^3)	0.36	14.21	236.81	0.00	0.00	226.7
f_d	1	1	11	0	0	11
f_s	1	1	1	0	0	1
R_t	99.74	99.74	97.14	100.00	100.00	97.14
R_v	99.90	99.80	98.95	100.00	100.00	99.00
φ	1.00	1.00	0.09	-	-	0.09
η	1.000	1.000	0.330	0.000	0.000	0.305
λ	0.000	0.000	0.390	1.000	1.000	0.395
λ_G	0.292			0.546		



(a) (b)

Figure 8: The optimised rule curves at Ubonratana (a) using SGA (b) using DGA

Conclusion

This study has developed the optimised rule curves of the Ubonratana reservoir in north-eastern Thailand using a new approach of Dynamic Genetic Algorithm (DGA). The standard genetic algorithm (SGA) has been improved by search space modification using DGA in which boundaries are continuously updated by modified search space reduction technique (SSRT). The search space is then focused around the area of the optimal solution, hence speeding up the convergence process and improving the precision of solutions. Comparing the performance of the reservoir when operated with the rule curves optimised with SGA and DGA showed that the DGA curves were far superior to the SGA curves. In particular the evaluated sustainability indices showed that the DGA was better than the SGA, for the individual water supply categories as well as their aggregation. A further attribute of the DGA is its speed at arriving at the global optimum. For example, recorded computational times for the DGA were on average about half of those required by a standard algorithm solving the same problem.

References

Adeloye AJ (2012) Hydrological sizing of water supply reservoir. In *Encyclopaedia of lakes and reservoirs* (Bengtsson, L, Herschy, RW, and Fairbridge, RW (eds)). Springer; Dordrecht, 346-355.

Chang FJ, Chen L and Chang LC (2005) Optimizing the reservoir operating rule curves by genetic algorithms. *Hydrological Processes,* 19(11), 2277-2289.

Chiamsathit C, Adeloye AJ and Soundharajan B (2014) Genetic algorithms optimization of hedging rules for operation of the multi-purpose Ubonratana reservoir in Thailand. *Proc. IAHS, 364, 507-512*, doi:10.5194/piahs-364-507-2014.

EGAT (2002) Improved Rule Curve. Procedure of the Ubonratana reservoir operation: Electricity Generating Authority of Thailand (EGAT) in the Ubonratana dam, Khon Kaen, Thailand.

Fennessey NM (1995) sensitivity of reservoir-yield estimates to model time step surface-moisture fluxes. *J. Wat. Resour. Plan. Managt., ASCE*, 121(4), 310-317.

Hashimoto T, Stedinger JR and Loucks DP (1982) Reliability, resilience and vulnerability criteria for water resources system performance evaluation. *Water Resources Research,* 18(1): 14–20.

Liu J (2012) An adaptive boundary genetic algorithm for continuous optimization problem. *Emerging research in artificial intelligence and computational intelligence*, International conference, AICI 2012, 357-362

McMahon TA and Adeloye AJ (2005) Water resources yield. Water Resources Publications, LLC, Colorado, USA.

McMahon TA, Adeloye AJ and Zhou SL (2006) Understanding performance measures of reservoirs. *Journal of Hydrology,* 324(1–4): 359–382.

Metkar SJ and Kulkarni AJ (2013) Boundary searching genetic algorithm: a multi-objective approach for constrained problems. In *Advances in intelligent systems and soft computing* (Satapathy SC, Biswal BN, Udgata SK (eds)). Springer, 269-276

Michalewicz Z (1992) Genetic algorithms + data structures = evolution programs, Springer, New York.

Ndiritu JG and Daniell TM (2001) An improved genetic algorithm for rainfall-runoff model calibration and function optimization. *Mathematical and Computer Modelling*, 33, 695-706

Purohit GN, Sherry AM and Saraswat M (2013) Optimization of function by using a new MATLAB based genetic algorithm procedure. *International Journal of Computer Applications*, 6(15)

Raff LM, Komanduri R, Hagan M and Bukkapatnam STS (2012) Genetic algorithm (GA) and internal energy transfer calculation using neural network (NN) methods. In Neural network in chemical reaction dynamics. Oxford University press, Inc., 165-167.

Roeva O, Fidanova S and Paprzycki M (2013) Influence of the population size on the genetic algorithm performance in case of the cultivation process modelling. *Proceeding of the 2013 federated conference on computer science and information systems*, 371-376

Sandoval-Soils, S, Mckinney, D.C. and Loucks, D.P. (2011) 'Sustainability index for water resources planning and management', *Water Resources Planning and Management*, ASCE, 137(5), 381-389.

Senthil kumar AR, Goyal MK, Ojha CSP, Singh RD, Swamee PK and Nema RK (2012) Application of ANN, Fuzzy Logic and Decision Tree Algorithms for the Development of Reservoir Operating Rules. Water Resour Manage (2013), 27, 911–925.

Wu ZY and Simpson AR (2002) A self-adaptive boundary search genetic algorithm and its application to water distribution systems. *Journal of hydraulic research*, 40(2), 191-203.