

Multiobjective water resources systems analysis using genetic algorithms – application to Chou-Shui River Basin, Taiwan

J.-T. Kuo*, W.-C. Cheng** and L. Chen***

* Department of Civil Engineering and Hydrotech Research Institute, National Taiwan University, Taipei, Chinese Taiwan (E-mail: kuoj@ccms.ntu.edu.tw)

** Former graduate student, Department of Civil Engineering, National Taiwan University, Taipei, Chinese Taiwan

*** Department of Civil Engineering, Chung Hua University, Hsin Chu, Chinese Taiwan 30067 (E-mail: lichen@chu.edu.tw)

Abstract Multipurpose operation is adopted by most reservoirs in Taiwan in order to maximize the benefits of power generation, water supply, irrigation and recreational purposes. A multiobjective approach can be used to obtain trade-off curves among these multipurpose targets. The weighting method, in which different weighting factors are used for different purposes, was used in this research work. In Taiwan, most major reservoirs are operated by rule curves. Genetic algorithms with characteristics of artificial intelligence were applied to obtain the optimal rule curves of the multireservoir system under multipurpose operation in Chou-Shui River Basin in central Taiwan. The model results reveal that different shapes of rule curves under different weighting factors on targets can be efficiently obtained by genetic algorithms. Pareto optimal solutions for a trade-off between water supply and hydropower were obtained and analyzed.

Keywords Chou-Shui River Basin; genetic algorithms; multiobjective programming; rule curves

Introduction

The study of genetic algorithms (GAs) was originated by Holland (1975). GAs have been developed to become a powerful optimization approach. In the theory of GAs, Brindle (1981) improved the roulette wheel selection; Wright (1991) and Eshelman *et al.* (1992) developed real-value coding to solve related problems. GAs were applied to water resources systems by Goldberg and Kuo (1987) for pipeline optimization. Wang (1991) applied GAs to the calibration of a conceptual rainfall-runoff model. McKinney and Lin (1994), Ritzel *et al.* (1994), and Cieniawski *et al.* (1995) solved ground-water problems with GAs.

So far, there have been very few applications of GAs to reservoir systems optimization. East and Hall (1994) applied GAs to a four-reservoir problem; the objective was to maximize the benefits from power generation, irrigation and water supply. Fahmy *et al.* (1994) also applied GAs to a reservoir system, and compared the performance of the GAs with dynamic programming. Oliveira and Loucks (1997) used GAs to evaluate operating rules for a multi-reservoir system. Chang and Chen (1998) applied a GA to solve a flood control operation problem. Sharif and Wardlaw (2000) used GAs to solve a multi-reservoir system optimization problem in Brantas Basin located in Indonesia.

This research uses GAs for obtaining optimal rule curves in a reservoir system operation and finds the Pareto optimal solutions for water supply and hydropower generation in Chou-Shui River Basin in central Taiwan.

Genetic algorithms

Characteristics of GAs

The need for solving optimization problems arises in almost every field. Genetic

algorithms (GAs) is one of the most promising techniques in that domain and has received a great deal of attention regarding its potential for optimizing complex systems. The GA is essentially a Darwinian natural selection, which combines an artificial survival of the fittest with natural genetic operators (Holland, 1975). Through the genetic evolution method, an optimal solution can be found and represented by the final winner of the genetic evolution.

The genetic algorithm is an iterative procedure maintaining a population of individuals that are candidate solutions for a specific domain. During each generation, the individuals in the current population are rated for their effective evaluations, and a new population of candidate solutions is formed using specific genetic operators such as reproduction, crossover, and mutation (Grefenstette, 1986). The cycle of evolution is repeated until a desired termination criterion is reached. This criterion can also be set by the number of evolution cycles (computational runs), the variation of individuals between different generations, or a pre-defined value of fitness.

The major differences between GAs and other search methods can be summarized as follows.

- GAs are easy to implement in parallel architecture;
- GAs address multi-objective problems;
- GAs are capable of handling problem with constraints; and
- GAs can solve multi-modal, non-differentiable, non-continuous or even NP-complete problems.

Representation and operators

The common method of applying GAs to real-parameter problems is to encode each parameter as a bit string using a standard binary coding. A GA is composed of three operators: reproduction, crossover and mutation.

Coding scheme. Individuals A are represented as bit strings of fixed length L , i.e. $A \in B^L$ where $B = \{0,1\}$. The bit strings for the parameters are concatenated together to give a single bit string (or “chromosome”) which represents the entire vector of parameters. In biological terminology, each bit position corresponds to a gene of the chromosome, and each bit value corresponds to an allele. If a single parameter x_i has lower and upper bounds a_i and b_i respectively, then the standard way of binary coding x_i using n bits is to let real value $a_i + m(b_i - a_i)/2^n$ correspond to the standard binary code for an integer m in the range $0 \leq m < 2^n$.

Reproduction or selection. Reproduction is a process in which individual strings are set according to their objective function values, f (biologists call this function the fitness function). Producing strings according to their fitness values means that strings with a higher value have a higher probability of contributing one or more offspring in the next generation. This operator, of course, is an artificial version of natural selection. It is Darwinian survival of the fittest among string creatures. The reproduction operator may be implemented in an algorithmic form in a number of ways, but tournament selection will be used in this paper. Tournament selection is computationally more efficient and more amenable to parallel implementation (Goldberg and Deb, 1990). Two individuals are chosen randomly from the population. A random number u is then chosen between 0 and 1. If $u < v$ (where v is a parameter, for example 0.7), the fitter of the two individuals is selected to be a parent; otherwise the less fit individual is selected.

Crossover or recombination. After reproduction, the algorithm uses a crossover operator that exchanges arbitrary substrings between two individuals with probability P_c . Goldberg

(1989) and Michalewicz (1992) described the following methods of crossover: (1) one-point crossover; (2) two-point crossover; and (3) uniform crossover. One-point crossover may proceed in two steps. First, members of the newly reproduced strings in the mating pool are mated at random. Second, each pair of strings undergoes crossing over by the procedure described below. An integer position k along the string is selected uniformly at random between 1 and the string length less one $[1, L-1]$. Two new strings are created by swapping all characters between positions $k+1$ and L inclusively. In two point crossover, genetic material between two positions chosen at random along the length of the chromosomes is exchanged. Uniform crossover operates on individual genes of the selected chromosomes, rather than on blocks of genetic material, and each gene is considered in turn for crossover or exchange (Wardlaw and Sharif, 2000). In this paper, uniform crossover will be used to recombine two parent chromosomes.

Mutation. In a GA, mutation is the occasional random alteration, with small probability (P_m), of the value of a string position. It changes 1 to 0 and vice versa. By itself, mutation is a random walk through the string space. When used sparingly with reproduction and crossover, it is an insurance policy against genetic drift which will lose important notions.

Parameters setting

When defining a GA one needs to choose its parameters, such as the population size, probability of crossover and mutation. The values of these parameters greatly determine whether the algorithm will find a near-optimum solution and whether it will find such solution efficiently (Eiben *et al.*, 1999). According to De Jong (1975), the population size N is set to be 100, whereas the crossover rate P_c equals 0.5, since the tournament selection size s is 2 (Reed *et al.*, 2000). The mutation rate P_m is set as the inverse of the population size, $P_m = 1/N = 0.01$ (Reed *et al.*, 2000).

Constraints

When dealing with constrained optimization problems, penalty functions are often used. A common technique is the method of penalties (Zhang and Muhlembein, 1995). The main reason for its wide-spread use is that it is the simplest technique to implement: it requires only the straightforward modification of the fitness function fit as follows:

$$fit(\vec{x}) = f(\vec{x}) + W \cdot penalty(\vec{x}) \quad (1)$$

where f is the objective function, and penalty is zero if no violation occurs, and is positive otherwise (for minimization problems).

Multiobjective programming by genetic algorithms

Schaffer (1985) created a loop around traditional procedure so that selection is repeated for each criterion or objective to fill up a portion of the mating pool called VEGA (Vector Evaluated Genetic Algorithm). Srinivas and Deb (1995) developed NSGA (Nondominated Sorting Genetic Algorithm) based on ranking which was suggested by Goldberg (1989), and then brought into the concept of elitist to develop ENGA (Elitist Nondominated Sorting Genetic Algorithm).

In this paper, the weighting method is used for the multiobjective optimization. Recently, Murata *et al.* (1996) used a weighted sum of the objective functions being maximized to produce Pareto optimal solutions. Comparing the weighting method to VEGA, Bachi (1999) reported that the weighting method performs better than VEGA.

Case study

Chou-Shui River Basin

The Chou-Shui River Basin, located in the central region of Taiwan, has the second largest catchment area (3,155km²) in the country. The existing water resource systems in the basin are comprised of two reservoirs (Wu-Shuh and Sun Moon Lake), three diversion dams (Wan-Ta, Wu-Chieh and Chi-Chi), four conventional hydropower plants (Wan-Ta, Ta-Kuan I, Chu-Kung, and Shui-Li), and two pumped storage power plants (Ta-Kuan II and Ming-Tan). The purposes of reservoir operation include hydropower generation, irrigation, public water supply and industrial water supply.

Objective function

Power generation. Power generation is one of the main targets, so the total power generation (kWh) is included in the objective function. In the model, we neglected short-term power generation demand, frequency modulation and regulated voltage because of their random property in a small time scale. The power constant changes with water level in order to calculate the power generation for Wan-Ta and Ta-Kuan power plants. A fixed power generation constant is used in other power plants.

Water supply. Minimization of ten-day shortage index was adopted for the objective function in order to meet the goal of stable water supply:

$$SI = \frac{100}{N} \sum \left(\frac{S}{D}\right)^2 \tag{2}$$

where *SI* = ten-day shortage index, *S* = ten-day shortage, *D* = ten-day water demand, and *N* = number of total stages.

This type of objective function for *SI* tends to make an even distribution of shortage in time domain, and reduce the extreme situation. In theory, the optimized operation will try to make the objective function equal to zero, which means no shortage of water.

Combining power generation and water supply

In order to maximize power generation and minimize *SI* at the same time, the approach of multi-objective programming is applied. Normalizing both objective functions is used:

$$\text{Maximize } \left\{ W_P \left(\frac{Power - MinP}{MaxP - MinP} \right) - W_I \left(\frac{Index - MinI}{MaxI - MinI} \right) \right\} \tag{3}$$

where *W_P* and *W_I* = weighting factors for hydropower generation (*P*) and water shortage (*I*); Max and Min = maximum and minimum values for *P* and *I*, respectively. Power or *P* = hydropower, and Index or *I* = shortage index.

Modified objective function

In this paper, a penalty coefficient *R* is additionally applied for the objective function, and it can improve *GAs* in searching for a reasonable solution quickly in a feasible region. They are defined as Eqs (4) to (9):

Power generation

$$R = 1 - \frac{C}{N} \tag{4}$$

$$C = \sum_{i=1}^N \frac{RC(i) - D(i)}{RC(i) - MINRC} \quad \text{for } RC(i) > D(i) \tag{5}$$

where N = number of parameters, $RC(i)$ = real rule curve, $D(i)$ = rule curve represented by chromosomes, and $MINRC$ = lower limit of reservoir.

Water supply

$$R = 1 - \frac{C}{N} \tag{6}$$

$$C = \sum_{i=1}^N \frac{RC(i) - D(i)}{RC(i) - MINRC} \quad \text{for } RC(i) < D(i) \tag{7}$$

Combining power generation and water supply

$$R = 1 - \left(\frac{C}{N}\right)^2 \tag{8}$$

$$C = \sum_{i=1}^N \frac{RC(i) - D(i)}{RC(i) - MINRC} \quad \text{for } RC(i) > D(i) \tag{9}$$

Results and discussion

Two main purposes, hydropower generation and water supply, were considered to implement the multiobjective optimization analysis. Six cases were analyzed and discussed. Case I: power generation is the main target; Case II: water supply is the main target; Case III: weighting factors are 50% for power generation and 50% for water supply; Case IV: 20% for power generation and 80% for water supply; Case V: 80% for power generation and 20% for water supply; and Case VI: 90% for power generation and 10% for water supply. Figures 1 and 2 are the optimal rule curves for Case I and its comparison to presently used rule curves. Figures 3 and 4 are the results for Case III.

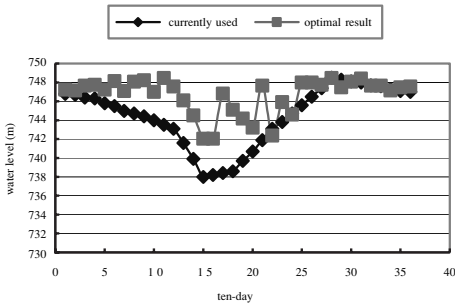


Figure 1 Comparison of currently used and optimal rule curves in Sun Moon Lake for Case I

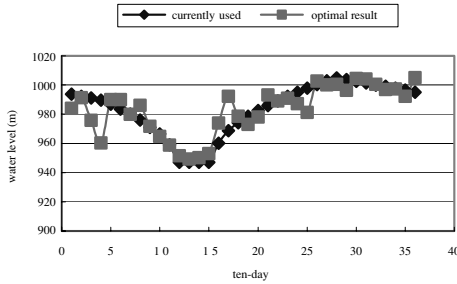


Figure 2 Comparison of currently used and optimal rule curves in Wu-Shieh Reservoir for Case I

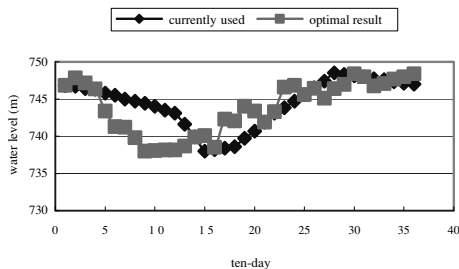


Figure 3 Comparison of currently used and optimal rule curves in Sun Moon Lake for Case III

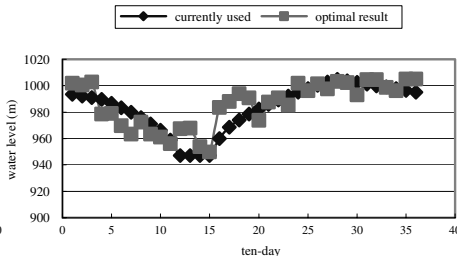


Figure 4 Comparison of currently used and optimal rule curves in Wu-Shieh Reservoir for Case III

Table 1 Optimal results of Case I to Case VI

Alternative	Ten-day shortage index	Average power generation potential (10 ⁶ kwh)
Case I	4.580	7724.776
Case II	2.973	7702.595
Case III	3.056	7705.075
Case IV	3.026	7703.504
Case V	3.133	7711.795
Case VI	3.905	7717.856

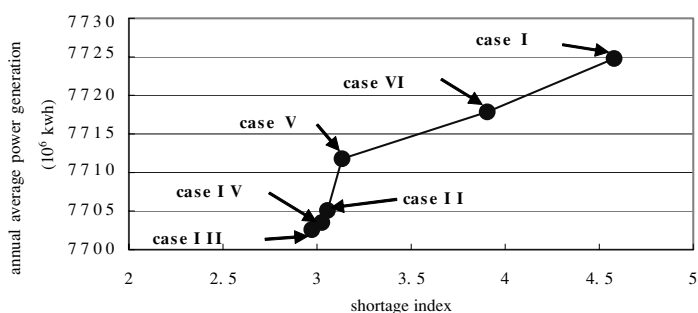


Figure 5 Pareto optimal solution for six cases

The result reveals that varying shapes of rule curves are obtained for different weighting factors. Finally, from the above studies of the six cases, we can obtain the Pareto optimal solution (see Table 1 and Figure 5).

Results show that water supply for the downstream area is influenced significantly by the level of rule curve during February to May, owing to the high water requirement for rice paddies. That is, under the circumstance of increasing release amount or at lower values of rule curves, the situation of shortage can be efficiently improved. On the other hand, when higher weighting factors for power generation are considered, higher levels of rule curves will occur. The same trend can be observed during the last half year. When a higher weighting factor for water release is considered, the level of rule curves is contrary to the former.

Conclusions

Pareto optimal solution was obtained from the optimization process by GAs under different weighting factors for power generation and water supply. If we want to increase the total power generation or decrease the shortage of water, lower values of rule curves can achieve the goal according to the result of optimization during February to May. Although the optimal results show that the shape of the rule curve is not smooth, it does show the trends of optimal rule curves in different reservoirs. In the future, for a complete study, benefits of recreation can be taken into account. The ground water system can be also considered in conjunctive use with surface water. Different penalty functions can be analyzed and compared to find the improved optimal solution.

References

- Bachi, T.P. (1999). *Multiobjective Scheduling by Genetic Algorithms*. Kluwer Academic Publishers.
- Brindle, A. (1981). *Genetic Algorithms for Function Optimization*. Unpublished doctoral dissertation. University of Alberta, Edmonton, Canada.
- Chang, F.J. and Chen, Li (1998). Real-code genetic algorithm for rule-based flood control reservoir management. *Water Resour. Management*, **12**, 185–198.

- Cieniawski, S.E., Eheart, J.W. and Ranjithan, S. (1995). Using genetic algorithms to solve a multiobjective groundwater monitoring problem. *Water Resour. Res.*, **31**(2), 399–409.
- De Jong, K.A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD dissertation, University of Michigan, Ann Arbor, Mich.
- East, V. and Hall, M.J. (1994). Water resources system optimization using genetic algorithms. *Hydroinformatics '94, Proc., 1st Int. Conf. on Hydroinformatics*, Balkema, Rotterdam, The Netherlands, 225–231.
- Eiben, A.E., Hinterding, R. and Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Trans. on Evolutionary Computation*, **3**(2), 124–141.
- Eshelman, L.J., Schaffer and David, J. (1992). Real-coded genetic algorithms and interval – schemata. *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 187–202.
- Fahmy, H.S., King, J.P., Wentzel, M.W. and Seton, J.A. (1994). Economic optimization of river management using genetic algorithms. Paper No. 943034, *ASCE Int. Summer Meeting, Am. Soc. of Agricultural Engrs.*, St. Joseph, Mich.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass.
- Goldberg, D.E. and Deb, K. (1990). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, ed. G.E. Rawlins, Morgan Kaufman, San Mateo, Calif. pp. 63–93.
- Goldberg, D.E. and Kuo, C.H. (1987). Genetic algorithms in pipeline optimization. *J. Comp. In Civ. Engrg.*, ASCE, **1**(2), 128–141.
- Grefenstette, J.J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on System, Man, and Cybernetics*, **SMC-16**(1), 122–128.
- Holland, J.H. (1975). *Adaptation in Nature and Artificial Systems*. MIT Press, Cambridge.
- McKinney, D.C. and Lin, M.D. (1994). Genetic algorithms solution of groundwater management models. *Water Resources Research*, **30**(6), 1897–1906.
- Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evaluation Programs*. Springer, New York.
- Murata, T., Ishibuchi, H. and Tanaka, H. (1996). Multiobjective genetic algorithm for flowshop scheduling problem. *Computers Ind. Engng.*, **30**(4), 1061–1071.
- Oliveira, R. and Loucks, D.P. (1997). Operating rules for multireservoir systems. *Water Resour. Res.*, **33**(4), 839–852.
- Reed, P., Minsker, B. and Goldberg, D.E. (2000). Designing a competent simple genetic algorithm for search and optimization. *Water Resources Research*, **36**(12), 3757–3761.
- Ritzel, B.J., Eheart, J.W. and Ranjithan, S. (1994). Using genetic algorithms to solve a multiple objective groundwater pollution problem. *Water Resour. Res.*, **30**(5), 1589–1603.
- Schaffer, J.D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. *The First International Conference on Genetic Algorithms and Their Application*, 93–100.
- Sharif, M. and Wardlaw, R. (1999). Evaluation of genetic algorithms for optimal reservoir system operation. *Journal of Water Resources Planning and Management*, ASCE, **125**(1).
- Srinivas, N. and Deb, K. (1995). Multiobjective optimization using nondominated sorting genetic algorithm. MIT Press, *Evolutionary Computations*, **2**(3), 221–248.
- Wang, Q.J. (1991). The genetic algorithms and its applications to calibrating conceptual rainfall-runoff models. *Water Resource Res.*, **27**(9), 2467–2471.
- Wardlaw, R. and Sharif, M. (2000). Multireservoirs systems optimization using genetic algorithms: case study. *Journal of Computing in Civil Engineering*, **14**(4).
- Wright, A. (1991). Genetic algorithms for real parameter optimization. In *Foundations of Genetic Algorithms*, G.J.E. Rawlins (editor), Morgan Kaufmann, San Mateo, CA, 205–218.
- Zhang, B. and Muhlenbein, H. (1995). Balancing accuracy and parsimony in genetic programming. *Evol. Comput.*, **3**(3), 17–38.