

Reservoir Operating by Ant Colony Optimization for Continuous Domains (ACO_R) Case Study: Dez Reservoir

A. B. Dariane, and A. M. Moradi

Abstract—A direct search approach to determine optimal reservoir operating is proposed with ant colony optimization for continuous domains (ACO_R). The model is applied to a system of single reservoir to determine the optimum releases during 42 years of monthly steps. A disadvantage of ant colony based methods and the ACO_R in particular, refers to great amount of computer run time consumption. In this study a highly effective procedure for decreasing run time has been developed. The results are compared to those of a GA based model.

Keywords—Ant colony optimization, continuous, meta-heuristics, reservoir, decreasing run time, genetic algorithm.

I. INTRODUCTION

META-HEURISTIC methods are a sub-class of Monte Carlo algorithms that often work population-based or physical process as heuristic function [14]. Some examples of well-known meta-heuristic methods include Simulated annealing (SA), Tabu Search (TS), Genetic Algorithms (GA), Evolutionary Algorithms (EA), Evolution Strategy (ES), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO).

In this paper, formulation and application of ACO algorithm for continuous domains in water resources management has been studied. ACO algorithms are basically developed for discrete optimization and hence in continuous problems require discretization of the domains. ACO_R is believed to be a way to extend a generic ACO to continuous domains without any major conceptual change to its structure [13]. In ACO_R the continuous problem is tackled directly which eliminates the need for discretization.

The applications of ACO algorithms to water resources problems are very few. For instance, Abbaspour *et al.* used an ACO algorithm to estimate hydraulic parameters of unsaturated soil [1]. Maier *et al.* applied ACO algorithms to find an optimal solution to a water distribution system, pointing out that the ACO algorithms may achieve an appealing alternative to genetic algorithms for the optimum design of water distribution systems [6]. Nagesh Kumar and Janga Reddy used ACO to derive operating policies for a multi-purpose reservoir system and compared it with real

coded Genetic Algorithm. They concluded that ACO model outperforms GA model, especially in the case of long-time horizon reservoir operation [10]. Mortazavi Naeini applied ACO for optimization of a single reservoir operation by discretizing the reservoir volume into several intervals [8]. Shouju Li *et al.* used hybrid ant colony system with simulated annealing for parameter estimation in a groundwater problem. They mentioned that hybrid ant colony system is a global search and can find parameter set in a stable way [12].

In this paper we demonstrate the development and application of ACO_R for optimization of the Dez reservoir operation during 42 years of monthly steps (consisting of 12×42 decision variables). The reservoir is located in southwest Iran and is a major source of hydropower production and water supply for agriculture and domestic water needs.

II. ANT COLONY OPTIMIZATION

ACO algorithms were first proposed by Dorigo [3]. The inspiring source of ACOs is the foraging behavior of real ants. When searching for food, ants explore the area surrounding their nest in a random way. As soon as an ant discovers a food source, it evaluates and transfers some food back to the nest. At this time, the ant deposits a pheromone trail on the ground. The pheromone deposited, guides other ants to the food source [3],[13]. The ACO meta-heuristic is shown and explained in the following [3],[13].

ACO meta-heuristic algorithm

```

Set parameters, initialize pheromone trials
While (termination conditions not met) do
  Construct_Ants_Solutions
  Update_Pheromone
  Apply_local_Search      %optional
end

```

The algorithm starts with determination of the parameters and initializes pheromones on the trials. After that, these ACO algorithms iterate through a main loop, in which all of the ants construct solutions [3],[13]. Any movement from *i* to *j* is made using a probabilistic approach as described by (1).

$$P^k(c_{ij}) = \begin{cases} \frac{\tau_{ij}^\alpha \eta(c_{ij})^\beta}{\sum_{c_{il} \in N_i^k} \tau_{il}^\alpha \eta(c_{il})^\beta}, & \text{if } \forall c_{ij} \in N_i^k \\ 0, & \text{if } c_{ij} \notin N_i^k \end{cases} \quad (1)$$

A. B. Dariane is with the Khaje Nasir Toosi University of Technology, Civil Engineering Department, Tehran, Iran (e-mail: dariania@yahoo.com).

A. M. Moradi is with the Khaje Nasir Toosi University of Technology, Water Resources Management, Civil Engineering Department, Tehran, Iran (e-mail: amirmorady@gmail.com).

where $\eta(\cdot)$ is a heuristic value that assigns at each construction step to each feasible solution element $c_{ij} \in N_i^k$, α and β are two positive parameters which determine the relation influence of the pheromone information and heuristic information, N_i^k is value from a finite set of n elements D_i that ant k has not selected yet, and τ_{ij} is the pheromone value associated with element c_{ij} . τ_{ij} is explained as follows [3],[13].

$$\tau_{ij}^{t+1} = \begin{cases} (1-\rho) \cdot \tau_{ij}^t + \rho \cdot \sum_{k=1}^m \Delta \tau_{ij}^{t,k} & \text{if } \tau_{ij} \in s_{ch} \\ (1-\rho) \cdot \tau_{ij}^t & , \text{otherwise} \end{cases} \quad (2)$$

where $0 < \rho \leq 1$ is the evaporation rate. The parameter ρ is used to avoid too rapid convergence of the algorithm. It enables the algorithm to forget bad decisions previously taken. $\Delta \tau_{ij}^{t,k}$ is the amount of pheromone ant k deposits on the elements it has chosen. s_{ch} is the set of chosen good solution.

III. ACO_R

As mentioned in section II, in ACO at each construction step, ants make a probabilistic choice according to Eq(1). The probabilities associated make up a discrete probability distribution. But the fundamental idea in ACO_R is the change from using a discrete probability distribution to using a continuous one, that is, a probability density function (PDF)[13].

For this purpose, one of the most popular functions, i.e. the Gaussian function, is used. However, a single Gaussian function cannot describe a situation where two disjoint areas of the search space are promising, as it only has one maximum [13]. Due to this fact, Reference [13] defined a Gaussian kernel as a weighted sum of several one-dimensional Gaussian functions $g_l^i(x)$, and denote it as $G^i(x)$:

$$G^i(x) = \sum_{l=1}^k \omega_l g_l^i(x) = \sum_{l=1}^k \omega_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^i{}^2}} \quad (3)$$

where ω is the vector of weights associated with the single Gaussian functions, μ^i is the vector of means, and σ^i is the vector of standard deviations [13].

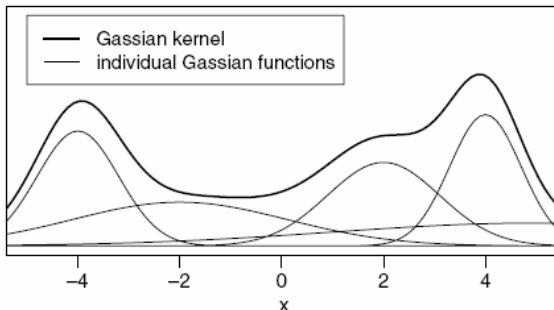


Fig. 1 Example of file single Gaussian functions in comparison to a Gaussian kernel

s_1	s_1^1	s_1^2	...	s_1^l	...	s_1^n	$f(s_1)$	ω_1
s_2	s_2^1	s_2^2	...	s_2^l	...	s_2^n	$f(s_2)$	ω_2

s_l	s_l^1	s_l^2	...	s_l^l	...	s_l^n	$f(s_l)$	ω_l

s_k	s_k^1	s_k^2	...	s_k^l	...	s_k^n	$f(s_k)$	ω_k

Fig. 2 The solution archive

Such a PDF allows an easy sampling and produces has much enhanced flexibility in the possible shape, in comparison to a single Gaussian function. Fig. 1 shows an example of such a Gaussian kernel PDF in comparison to several single Gaussian functions [13].

Solutions in ACO_R will be saved in an archive which is called *solution archive* (see Fig. 2). At the start of the algorithm, the solution archive is initialized by generating k solutions, often randomly.

In the archive k solutions $\{s_1, s_2, \dots, s_l, \dots, s_k\}$ with their objective function values $\{f(s_1), f(s_2), \dots, f(s_l), \dots, f(s_k)\}$ are kept. The i th variable of l th solution is hereby denoted by s_l^i . The solutions in the archive are kept according to their quality. Therefore, in a minimization problem will have:

$$f(s_1) \leq f(s_2) \leq \dots \leq f(s_l) \leq \dots \leq f(s_k)$$

where for each solution s_l has an associated weight ω based on the solution quality, therefore:

$$\omega_1 \geq \omega_2 \geq \dots \geq \omega_l \geq \dots \geq \omega_k$$

where the weight ω_l of the solution s_l is calculated according to the following formula:

$$\omega_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \quad (4)$$

According to (4) the weight is a value of the Gaussian function with argument l , mean 1.0 , and standard deviation qk , where q is a parameter of the algorithm. When q is small, the best-ranked solutions are preferred [13].

The second vector of the Gaussian kernel PDF which should be determined is μ^i . For each $G^i(x)$, the values of the i th variables of all the solution in the archive become the elements of the vector μ^i :

$$\mu^i = \{\mu_1^i, \dots, \mu_k^i\} = \{s_1^i, \dots, s_k^i\} \quad (5)$$

For determination of third vector σ^i , and regarding above remarks, suppose that an ant moves toward one of solutions such as s_l in solution archive by using a probability-oriented method such as Roulette Wheel by Holland [4], therefore, a solution whose rank is higher, has more chance to be chosen by the ants. Then, standard deviation between all i th decision

variable values shall be calculated in proportion with decision variable s_i^j that is shown by σ_i^j . For this purpose, by using a normal random generator such as Box-Muller method [2] with mean s_i^j and standard deviation σ_i^j shall be produced for i th decision variable and this process continues to the n th decision variable where finally the objective function and the function value $f(s_i)$ are calculated, for each ant. Finally, new solutions equal to the number of ants are produced and added to the archive. After collocation of all solutions, k solutions are reserved and the remaining solutions are deleted. In this direction, mentioned standard deviation shall be calculated as follows:

$$\sigma_i^j = \xi \sum_{e=1}^k \frac{|s_e^i - s_i^j|}{k-1}, \quad i = 1, 2, \dots, n \quad (6)$$

This formula is presented by Socha and Dorigo [13]. The positive parameter ξ is the same for all the dimensions and has an effect similar to the pheromone evaporation rate in ACO. The higher the value of ξ , makes decrease of the convergence speed of algorithm [13].

IV. CASE STUDY: DEZ RESERVOIR

Dez is the highest double-arched concrete dam in Iran and is located in southwest of the country. This reservoir is one of the multi-purpose reservoirs operated since 1962. It was constructed to supply water for agriculture of fertile plains between Andimeshk and Hafttapeh with an area of 125000 (Hectare). Meanwhile, with a plant capacity of 520 (MW),

hydropower production is another major purpose of this reservoir. Fig. 3 shows the reservoir location in the country.

The purpose of this paper is to find the monthly optimum releases for the 42 historical years (1956-1998).

The objective function is defined as minimization of cumulated operating loss function as follows:

$$Z_t = \begin{cases} (TD_t - R_t)^2 & \text{if } R_t < TD_t \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where Z_t = loss function and $t = 1, 2, \dots, 504$.

The problem is subjected to the following constrains:

$$S_{t+1} = S_t + Q_t - R_t - E_t \quad \forall t \quad (8a)$$

$$S_{\min} \leq S_t \leq S_{\max} \quad \forall t \quad (8b)$$

$$S_1 = S_{505} \quad (8c)$$

where S , Q , R , E , and TD are reservoir storage, inflow, release, evaporation, and monthly total demand, respectively. For a proper solution of the model using ACO_R algorithm, parameters should be estimated. Among them, two parameters of ξ, q have higher sensitiveness, and perhaps improper values of these, may conduct algorithm toward local optimums. The values of parameters for ACO_R are presented in Table I.

Higher numbers of ants selected, m , is important in reaching stable solutions, however, it also increases the computation time.



Fig. 3 Dez reservoir location in southwest Iran

TABLE I
THE REGULATED PARAMETERS FOR ACO_R

parameters	Symbol	Value
No. of ants	m	30
Speed of convergence	ξ	1.35
Locality of the search process	q	0.19
archive size	k	50

TABLE II
THE RESULTS OF 5 DIFFERENT RUNS FOR DEZ RESERVOIR PROBLEM BY ACO_R

Represents	value	unit
objective function	91.2×10^6	10^6 m^3
standard deviation	0.38×10^6	10^6 m^3
mean	91.56×10^6	10^6 m^3
run time	15840	second

Table II shows the results of the model application in Dez Reservoir obtained in 5 independent runs. It shows that the best value of objective function is 91.2×10^6 (mcm or 10^6 m^3), and the mean of 5 different runs equals to 91.56×10^6 (mcm or 10^6 m^3). The run time for each run is about 15840 (second) which is very high.

To overcome the run time problem a procedure was constructed which was inspired by the work of Lin-Kernighan on TSP problem [5] and Momtahan and Dariane [7] on Genetic algorithm. At first, the original problem called *big problem* is divided into some small parts called *small problems*. Then each of these *small problems* is solved separately by the ACO_R. Then, the collocated results of *small problems* are used as the initial solution in the *big problem*. In ACO_R, the best of solutions in the initial archive are kept intact and carried over the next step, while the remaining inferior solutions are replaced with uniform random solutions. In other words, all solutions existing in initial solution archive should not be used, because, produced initial solution archive may conduct algorithm to local optimums. The reason lays in the fact that each part of that archive has been optimized before. In that case, it leads to decrease in the standard deviation for the *ith* decision variable (see (6) in section III) and causes a decrease in the search space. The summary of results of this method has been shown in Table III.

TABLE III
THE RESULTS OF 10 DIFFERENT RUNS FOR DEZ RESERVOIR PROBLEM BY NEW ACO_R

Represents	value	unit
objective function	89.5×10^6	10^6 m^3
standard deviation	0.12×10^6	10^6 m^3
mean	89.60×10^6	10^6 m^3
run time	3000	second

Results as shown in Table III reveals that the best value of the objective function obtained in 3000 (second) is equal to 89.5×10^6 (mcm or 10^6 m^3) which shows an improved of 1.9% in comparison with the results of Table II. Meanwhile, the

standard deviation of values of the objective functions in 10 runs is 0.12×10^6 (mcm or 10^6 m^3) which shows a %70 decrease. More important, run time has decreased considerably by about fivefold.

Fig. 4 shows the trend of objective function with time. It takes about 775 (second) to prepare the initial solution archive. Calculations were stopped at 3000 (second) where no more improvement was observed in the value of objective function.

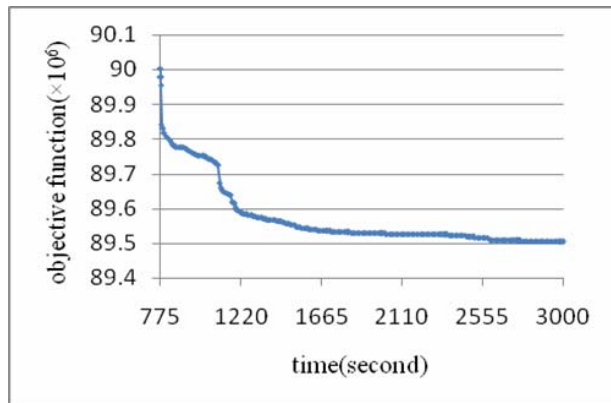


Fig. 4 Process of being converged the objective function to solve the new ACO_R model optimally

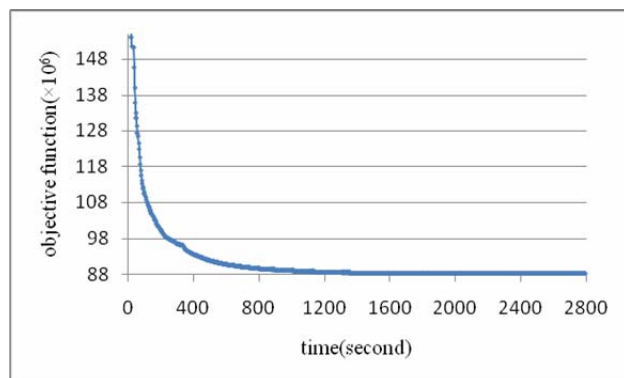


Fig. 5 Process of being converged the objective function to solve the GA model optimally

TABLE IV
THE RESULTS OF 5 DIFFERENT RUNS FOR DEZ RESERVOIR PROBLEM BY READ CODED GA

Represents	value	unit
objective function	88.2×10^6	10^6 m^3
standard deviation	0.1×10^6	10^6 m^3
mean	88.3×10^6	10^6 m^3
run time	2800	second

Furthermore, this problem was solved using a real coded GA. Operations of the GA model are truncation selection by Mühlenbein and Schlierkamp-Voosen [9],[11], extended intermediate recombination by Mühlenbein and Schlierkamp-Voosen [9],[11], and, mutation in evolutionary algorithms by Mühlenbein and Schlierkamp-Voosen [9],[11]. The results of GA model for Dez reservoir problem is represented in Table

IV. Also, the trend of objective function with time has been shown in Fig. 5. It shows that the best value of objective function reached in 2800 (second) is equal to 88.2×10^6 (mcm or 10^6 m^3). Therefore, the results of the proposed ACO_R method is very close to the real coded GA.

V. CONCLUSION

The modifications adopted in ACO_R indicate a substantial improvement in run time of the algorithm. It also slightly improves the value of the objective function. Comparison of the modified model with a GA model in optimization of Dez Reservoir shows that the new ACO_R is comparable to the well-known GA approach in solving relatively complex problems of water resources problems.

REFERENCES

- [1] Abbaspour, K. C., Schulin, R. & Van Genuchten, M. T. (2001). "Estimating unsaturated soil hydraulic parameters using ant colony optimization", *Adv. Water Resour.*, 24(8), 827-841.
- [2] Box, G.E.P., Muller, M.E., 1958. "A note on the generation of random normal deviates", *Annals of Mathematical Statistics* 29 (2), 610-611.
- [3] Dorigo, M., Stützle, T., 2004. "Ant colony optimization", MIT Press, Cambridge, MA, ch. 3.
- [4] Holland, J.H., 1975, "Adaptation in natural and artificial systems", Ann Arbor, MI: MIT Press.
- [5] Lin, S., Kernighan B. W.,(1971), "An effective heuristic algorithm for the traveling-salesman problem", *Oper. Res.* 21,498-516.
- [6] Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y. & Tan, C. L. (2003). "Ant colony optimization for design of water distribution systems", *J Water Resour Plan Manage.*, 129(3): 200-209.
- [7] Momtahn, Sh., and Dariane, A. B., (2007), "Direct search approaches using genetic algorithms for optimization of water reservoir operating policies", *Journal of water resources planning and management*. pp. 202-209.
- [8] Mortazavi Naeini, S. M., (2006), "Application of stochastic-heuristic methods in optimum water resources operation (in Persian)", Master thesis, K. N. Toosi University of Technology, Iran.
- [9] Mühlenbein, H., and Schlierkamp-Voosen, D., (1993), Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization. *Evolutionary Computation*, 1 (1), pp. 25-49.
- [10] Nagesh Kumar D., and Janga Reddy M., (2006), "Ant colony optimization for multi-purpose reservoir operation", *Water Resources Management* 20: 879-898.
- [11] Pohlheim, H., (2005), "GEATbx Introduction evolutionary algorithms: overview, methods, and operators", Documentation for: GEATbx version 3.7 (Genetic and Evolutionary Algorithm Toolbox for use with Method). [online]. Available: <http://www.geatbx.com>
- [12] Shouju Li, Yingxi Liu, and He Yu., (2006), "Parameter Estimation Approach in Groundwater Hydrology Using Hybrid Ant Colony System", Irwin (Eds.): ICIC 2006, LNBI 4115, pp. 182 - 191.
- [13] Socha, K., and Dorigo, M., (2006), "Ant colony optimization for continuous domains", *European Journal of Operational Research*, in press.
- [14] Weise, T., (2007), *Global Optimization Algorithms – Theory and Application*. [online]. ch. 1. Available: <http://www.it-weise.de>