Darbandikhan Reservoir Operation Optimization Using Ant Colony Optimization Algorithm

Mohammed Nasseh Mohammed, Bnar Faisal A. Daham, Kanar Shukr Mohammed

1Department of Surveying, College of Engineering, Salahaddin University-Erbil
2,3Department of Software and Informatics, College of Engineering, Salahaddin University-Erbil

ARTICLE INFO

Article History:
Received: 14/05/2017
Accepted: 12/04/2018
Published: 01/06/2018

Keywords:
Ant Colony Algorithm
Optimization
Reservoir operation

*Corresponding Author:
Email: mohamed_nasseh@yahoo.com
mohammed.mohammed@su.edu.krd

ABSTRACT

The importance of water resources management has unlimited scope, and from that importance comes the need for optimizing the operation of water reservoirs in terms of achieving hydropower generation demands, irrigation demands, and avoiding flood risks. There were many optimization techniques or methods have been conducted for that purpose.

In this paper we try to bring the operation of Darbandikhan reservoir (located 60 km southeast of Sulaimaniya city) to an optimum level using Ant Colony Optimization (ACO) technique. Thus, the objective is to find a monthly water release plan with least difference from the amount of water demand for that month.

This study covers the operation of the reservoir for one year, sampled into 12 monthly periods. Two methods for pheromone trail update - Iteration Best Path (IBP) and Iteration All Path (IAP) - have been used and tested in the ACO algorithm to find out how fit they are with the reservoir operation problem. Also two levels of reservoir storage discretization have been applied to the problem; 100 and 200 intervals.

Generally, the ACO algorithm showed a high performance in exploring the optimum solutions for the operation of Darbandikhan reservoir. The obtained results of the tests revealed that the IAP outperforms the IBP in finding the optimum solutions. While the tests of the two discretization resolutions showed that the 100 intervals is more efficient than the 200 intervals in getting better results with a certain number of iterations and artificial ants.

Keywords: Ant Colony Algorithm
Optimization
Reservoir operation

1. INTRODUCTION

Water resources planning is the management of water resources under a set of policies and regulations in order to achieve certain goals. Then when we speak of meeting goals, it is clear that we are not dealing with creation of perfection but rather with finding the best possible way of meeting aims within the limitations of resources, and being the meaning of optimization (Serafim, Lorenzo and Miodrag, 1991).

Large dams are usually built for different purposes such as urban water supply, power generation, agricultural, industrial, flood control, and environmental objectives. And recently, much research has been done to achieve certain objectives in optimal reservoir operation. The main research methodologies are about achieving the optimum level of release and optimal storage volume by considering the changes in inflow and needs (D. Nagesh and M. Janga, 2005; Rodrigo and Daniel, 1997).

Optimization techniques applied for finding optimal solutions were mostly limited by the complexities of non linear relationships in model formulation and by increase in the number of variables and constraints. As a
consequent, many heuristic and metaheuristic algorithms have been recently proposed, which can provide quite good and feasible results in an acceptable computation time (Dorian and Keith, 2005; M. R. Jalali, A. Afshar, and M. A. Mariño, 2006).

Ant Colony Optimization (ACO) Algorithm is a relatively new nature-inspired metaheuristic technique. The ACO is used in this paper as a key algorithm to find suitable and optimum operating policy for a multi-purpose reservoir system.

In the rest of the paper; a brief description about ACO and its procedure is presented first. Next, the details of the case study and model formulation for reservoir operation are explained. Finally the results are discussed, followed by the conclusions.

From literature, there is a great potential to apply ACO in the field of water resources problems. D. Nagesh and M. Janga (2005) employed ACO algorithm to derive operation policies for a multi-purpose reservoir system. M. R. Jalali et al. (2007) proposed ACO algorithms to optimize the operation of a water reservoir.

2. ACO ALGORITHM: GENERAL ASPECTS

Ant Colony Optimization (ACO) is a metaheuristic approach proposed by Dorigo (1992). ACO Algorithm is a nature-inspired algorithm and a member in the family of Swarm Intelligence Methods. It is based on the foraging behavior of ants in terms of finding the shortest possible path from their nest to the discovered food source (Marco and Christian, 2005).

Upon finding food, ants of some species return back to their nest while laying down a chemical substance called pheromone. When other ants find such a path, they will not keep scouting and travelling at random, but instead, they follow the trail.

Over time, the more time it takes for an ant to travel down the path and back again, the more pheromones have to evaporate. Whereas on short and more taken paths, pheromone density becomes higher. Thus, pheromone evaporation also has the advantage of avoiding the convergence to a locally optimal solution.

In ACO, a set of software data structures called artificial ants search for desirable solutions to a given optimization problem. To apply ACO, the problem is transformed into a weighted graph to find the best path. Those ants incrementally build solutions by moving on the graph (M. R. Jalali et al., 2007). The probability of an ant movement from state to another depends on the combination of two factors; first, the attractiveness of the move as computed by some heuristic indicating the desirability of that move, and second, the pheromone level or strength of the move, indicating how proficient it has been in the past to make that particular move (Christian, 2005).

![Fig. 1. A pseudo-code for ant colony optimization algorithm](image-url)

A probabilistic decision policy is used by the ants to direct their search towards the most interesting regions of the search space.

Let $\tau_{ij}(t)$ be the total pheromone deposited on path $ij$ at time $t$, and $\eta_{ij}(t)$ be the heuristic value of path $ij$ at time $t$, according to the measure of the objective function (R. Moeini and M. H. Afshar, 2009) (Marco, Gianni and Luca, 1999). Heuristic value is a measurement of objective function, which along with the pheromone ($\tau_{ij}$), will determine the transition probability from option $i$ to $j$, at time period $t$, as follows:
\[ P_{ij}(t) = \begin{cases} \frac{[\eta_{ij}(t)]^\alpha][\tau_{ij}(t)]^\beta}{\sum_{(i,j) \in \text{allowed}} [\eta_{ij}(t)]^\alpha[\tau_{ij}(t)]^\beta} & \text{if } j \in \text{Allowed} \\ 0 & \text{otherwise} \end{cases} \]  

(1)

Where \( \alpha \) and \( \beta \) are parameters that determine the relative importance of the pheromone trail against the heuristic value.

When all ants in the colony complete their tour, the pheromone is going to be updated as follows:

\[ \tau_{ij}(t)_{\text{iteration}} = \rho \cdot \tau_{ij}(t) + (1 - \rho) \cdot \Delta \tau_{ij} \quad 0 \leq \rho \leq 1 \]  

(2)

where \( \rho \) = evaporation rate.

One the other hand, there are several ways for finding \( \Delta \tau_{ij}(t) \). In this study, we used two approaches:

**First Method:** Iteration Best Path Pheromone Update (IBP)

In this method, the pheromone update is applied to the edges of only the best path chosen in each iteration.

\[ \Delta \tau_{ij}(t) = \begin{cases} \frac{1}{G^*_{ib}(m)} & \text{if } (i,j) \in \text{tour done by ant } kib \\ 0 & \text{otherwise} \end{cases} \]  

(3)

Where \( G^k_{ib}(m) \) is the value of the fitness function for the ant that took the best tour at iteration \( m \).

**Second Method:** Iteration All Path Pheromone Update (IAP)

In this method, the update of the pheromone is applied to the edges of every path generated during each iteration.

\[ \Delta \tau_{ij}(t) = \sum_{k=1}^{M} \tau m^k_{ij}(t) \]  

(4)

\[ \tau m^k_{ij}(t) = \begin{cases} \frac{1}{G^k(m)} & \text{if } (i,j) \in T^k(m) \\ 0 & \text{if } (i,j) \notin T^k(m) \end{cases} \]  

(5)

Where \( G^k(m) \) is the value of the fitness function for the tour \( T^k(m) \) taken by the \( k \)-th ant at iteration \( m \) (M. R. Jalali, A. Afshar, and M. A. Mariño, 2006).

3. **CASE STUDY DESCRIPTION**

Derbendikhan Dam is located on the Diyala Sirwan river approximately 65km south-east of Sulaimaniyah and 230km north-east of Baghdad, situated at latitude 35°6’46"N and longitude 45°42’24"E. It was constructed between 1956 and 1961 for several purposes including: hydroelectric power production, irrigation, and flood control.

Darbendikhan Dam with the 128m high embankment has a total design capacity at normal operating level (El. 485.00m) of 3,000 Mm³, of which 2,500 Mm³ is live storage and 500 Mm³ being dead storage (SMEC International Pty., 2006).

In this paper, the study has been accomplished on the operation of the dam in the year 2015, during which the live storage of the reservoir has a minimum and maximum range from 950Mm³ to 1500Mm³ of water, respectively. The monthly inflow and demand data of the dam have also been collected for 2015. So the focus will be on the storage and release amounts, then finding the optimal minimum demand deficit as an objective.

4.1 **Model Application**

It was observed that the storage capacity in 2015 was ranging between 950Mm³ and 1500Mm³, hence, in this study a range from 900Mm³ to 1700Mm³ has been taken and discretized into intervals. Two levels of resolution were applied on the intervals, 100 classes; 8Mm³ per each interval and 200 classes; 4Mm³ per each interval.

A predefined number of artificial ants as solution makers are placed on random classes in the first period. Then each of them is starting to decide which path to take. They select the transition edge from a storage class in time period \( t \) to another storage class in time period \( t+1 \), depending on two main factors:

First: The desirability (heuristic) of that transition edge, in terms of obtaining the minimum difference between the released and the amount of water demand at that period.

Second: The rank (pheromone trail) of that edge, which has been possibly gained from previous successful ants’ movement across that edge. There are different strategies for promoting the edges. Figure 2 shows the
discretized graph and an artificial ant when making decisions to find its path.

Fig. 2. Discretized graph and an artificial ant when making decisions to find its path

4.2 Heuristic Information

The heuristic information $\eta_{ij} (t)$ of the problem in this paper is determined by considering the criterion as minimum deficit.

$$
\eta_{ij}(t) = \frac{1}{R_{ij}(t)-D(t)} + c 
$$

for all $t = 1, 2, ..., NT$ (6)

Where $R_{ij}(t)$ = release at period $t$, depending on the initial and final storage volume at classes $i$ and $j$, respectively; $D(t)$ = demand of period $t$; and $c$ = a constant for irregularity avoiding (i.e. division by zero).

However, the continuity equation expressed below is used to determine $R_{ij}(t)$

$$
R_{ij}(t) = S_i - S_j + I(t) 
$$

Which is subject to the following constraints:

$$
S_{min} \leq S_i \leq S_{max} 
$$

$$
S_{min} \leq S_j \leq S_{max} 
$$

Where $S_i$ and $S_j$ = initial and final storage volumes in terms of classes $i$ and $j$, respectively. $I(t)$ = inflow to the reservoir at time period $t$. $S_{min}$ and $S_{max}$ = minimum and maximum storage allowed respectively.

4.3 Fitness Function

The goodness of the generated solutions is measured according to a defined fitness function. In this paper, the minimum value of the total square deviation (TSD) has been taken as the objective function:

$$
TSD^k = \sum_{t=1}^{NT} \left(\frac{(R^k(t) - D(t))}{D(t)}\right)^2 
$$

(10)

Where $R^k(t)$ = release at period $t$ recommended by ant $k$, and $NT$ = total number of periods.

5. TESTS AND RESULTS

For testing this model, several approaches and considerations were conducted. First point to be mentioned is that Java Programming Language is used in writing the source code of this model. Then, two phases of tests have been carried out including Parameters Tune-Up and Ants-Iterations tests.

Furthermore, both of the strategies of pheromone trail update were implemented at each level of the mentioned tests. Those strategies are IBP (promotes the edges of only the best path per iteration) and IAP (promotes the edges of every taken path per iteration).

5.1 Parameters Tune-Up

Hereby, the tests have been accomplished through tuning the main parameters used in ACOA, namely, $\alpha$ (determines the importance of the pheromone trail), $\beta$ (determines the strength of the heuristic value), and $\rho$ (determines the amount of pheromone increment or evaporation). Those tests have been conducted on only the 100-Class model with the number of ants and iterations being fixed to 5 and 500 respectively. Whereas the parameter were tuned up within the following ranges ($\alpha = [1, 2, 3, 4]$, $\beta = [1, 2, 3, 4]$, $\rho = [0.1, 0.5, 0.9]$). The results of each set of parameters as shown in Table 1 have been acquired through 10 runs of the algorithm.

As it could be observed from Table 1, the model of 100 classes provided optimum results for the IBP method of pheromone update when the parameters were tuned to ($\alpha = 3$, $\beta = 1$, and $\rho = 0.1$), resulting in a mean total square deviation of 0.105 units. While for the IAP
method of pheromone update, the optimum mean of total square deviation of 0.043 units was obtained when tuning the parameters to \((\alpha=3, \beta=1, \text{and } \rho=0.9)\). Figure 3 shows mean TSD results for parameters tune-up for both IBP and IAP methods in 100-Class model.

**Table 1.** TSD results for parameters tune-up for IBP and IAP methods in 100-Class model

| \(a, \beta, \rho\) | IBP Method | | | IAP Method | | |
|-----------------|-------------|-----------------|-----------------|-----------------|-----------------|
|                 | Best        | Worst           | Mean            | SD              | Best            | Worst           | Mean            | SD              |
| 4,1,0.1         | 0.053       | 0.353           | 0.135           | 0.066           | 0.05            | 0.113           | 0.08            | 0.021           |
| 4,1,0.5         | 0.075       | 0.273           | 0.183           | 0.057           | 0.043           | 0.108           | 0.081           | 0.018           |
| 4,1,0.9         | 0.105       | 0.622           | 0.349           | 0.191           | 0.043           | 0.073           | 0.032           | 0.013           |
| 3,1,0.1         | 0.066       | 0.142           | 0.105           | 0.027           | 0.051           | 0.191           | 0.127           | 0.041           |
| 3,1,0.5         | 0.042       | 0.236           | 0.157           | 0.056           | 0.048           | 0.147           | 0.104           | 0.033           |
| 3,1,0.9         | 0.105       | 0.611           | 0.287           | 0.172           | 0.026           | 0.096           | 0.043           | 0.010           |
| 2,1,0.1         | 0.077       | 0.211           | 0.142           | 0.039           | 0.083           | 0.239           | 0.171           | 0.051           |
| 2,1,0.5         | 0.056       | 0.205           | 0.147           | 0.049           | 0.133           | 0.245           | 0.182           | 0.035           |
| 2,1,0.9         | 0.167       | 0.611           | 0.317           | 0.154           | 0.036           | 0.112           | 0.073           | 0.028           |
| 1,1,0.1         | 0.051       | 0.177           | 0.119           | 0.044           | 0.141           | 0.428           | 0.232           | 0.102           |
| 1,1,0.5         | 0.114       | 0.254           | 0.166           | 0.045           | 0.091           | 0.322           | 0.226           | 0.066           |
| 1,1,0.9         | 0.167       | 0.617           | 0.362           | 0.144           | 0.037           | 0.112           | 0.076           | 0.029           |
| 1,2,0.1         | 0.087       | 0.177           | 0.141           | 0.021           | 0.132           | 0.166           | 0.332           | 0.181           |
| 1,2,0.5         | 0.079       | 0.327           | 0.17            | 0.066           | 0.152           | 0.401           | 0.303           | 0.099           |
| 1,2,0.9         | 0.167       | 0.498           | 0.313           | 0.139           | 0.073           | 0.166           | 0.119           | 0.028           |
| 1,3,0.1         | 0.066       | 0.258           | 0.128           | 0.058           | 0.137           | 0.462           | 0.318           | 0.136           |
| 1,3,0.5         | 0.123       | 0.823           | 0.236           | 0.062           | 0.026           | 0.284           | 0.201           | 0.054           |
| 1,3,0.9         | 0.167       | 0.617           | 0.351           | 0.156           | 0.091           | 0.139           | 0.114           | 0.016           |
| 1,4,0.1         | 0.054       | 0.177           | 0.109           | 0.042           | 0.167           | 0.472           | 0.282           | 0.104           |
| 1,4,0.5         | 0.043       | 0.251           | 0.146           | 0.063           | 0.091           | 0.544           | 0.288           | 0.144           |
| 1,4,0.9         | 0.167       | 0.61            | 0.32            | 0.165           | 0.111           | 0.21            | 0.179           | 0.027           |

In this phase, the models with both discretization resolutions 100-Class and 200-Class have been tested, and each type of them has been considered with IBP and IAP pheromone updating methods. However, the parameters \(a, \beta, \text{and } \rho\) were fixed upon the optimum tunes recorded in the previous test. The numbers of ants and iterations taken for this test were: Ants=[10, 20, 50], Iterations=[50, 100, 500]. The results of each combination of ants and iterations are shown in Table 2 and Table 3, where the best and worst results came out from 10 runs of each combination have been stated, in addition to the mean and standard deviation of the results acquired from those runs.

**Table 2.** TSD results for different number of ants and iterations using both IBP and IAP methods in 100-Class model

| Iterations | Status | Number of Ants (IBP) | Number of Ants (IAP) |
|------------|--------|----------------------|----------------------|
| 50         | Best   | 0.167                | 0.167                |
|            | Worst  | 0.765                | 0.472                |
|            | Mean   | 0.454                | 0.291                |
|            | SD     | 0.212                | 0.137                |
| 100        | Best   | 0.167                | 0.167                |
|            | Worst  | 0.615                | 0.472                |
|            | Mean   | 0.454                | 0.291                |
|            | SD     | 0.212                | 0.137                |
| 500        | Best   | 0.167                | 0.167                |
|            | Worst  | 0.472                | 0.472                |
|            | Mean   | 0.278                | 0.231                |
|            | SD     | 0.110                | 0.117                |

**Table 3.** TSD results for different number of ants and iterations using both IBP and IAP methods in 200-Class model

| Iterations | Status | Number of Ants (IBP) | Number of Ants (IAP) |
|------------|--------|----------------------|----------------------|
| 50         | Best   | 0.246                | 0.162                |
|            | Worst  | 0.844                | 0.406                |
|            | Mean   | 0.608                | 0.403                |
|            | SD     | 0.219                | 0.219                |
| 100        | Best   | 0.246                | 0.162                |
|            | Worst  | 0.605                | 0.348                |
|            | Mean   | 0.358                | 0.251                |
|            | SD     | 0.143                | 0.074                |
| 500        | Best   | 0.246                | 0.162                |
|            | Worst  | 0.471                | 0.348                |
|            | Mean   | 0.279                | 0.252                |
|            | SD     | 0.125                | 0.089                |

5.2 Ants-Iterations

In this test the performance of the algorithm has been measured in terms of the number of artificial ants that are generating solutions and the number of iterations or trials they make to find better solutions.

Fig. 3. TSD results for parameters tune-up for IBP and IAP methods in 100-Class model
Table 2 and Table 3 demonstrate that for a certain number of iterations, the more ants were used for searching the solutions the better mean total square deviation is acquired. That means the 50 ants find out better results than 10 ants. Also for a determined number of ants, better value for the mean total square deviation is obtained as the number of iterations used for searching the solutions is increased. This means 500 iterations bring better results than 50 iterations.

On the other hand, the results achieved by IAP method for pheromone update are better than those of IBP method, for a certain number of ants and iterations in both methods.

Also, one can notes that no significant improvements have been made in the results when 200-Class model is used instead of 100-Class model, especially with 500 iterations. Table 4 lists the amounts of water demand of Darbandikhan reservoir for 2015 in terms of 12 months with the optimized release (in million M$^3$) suggested by the four different approaches applied in this study. Also Figure 4 illustrates the content of Table 4, showing the performance of the four proposed models admitting how optimum results are acquired.

**Table 4.** Water demand for 2015 with the optimized suggestion of release in million M$^3$ using the four different approaches

| Demand | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 100-Class IBP | 166 | 73 | 79 | 99 | 127 | 245 | 230 | 92 | 64 | 114 | 75 |
| 100-Class IAP | 166 | 73 | 79 | 99 | 127 | 245 | 230 | 92 | 64 | 114 | 83 |
| 200-Class IBP | 166 | 73 | 79 | 99 | 127 | 245 | 230 | 92 | 64 | 114 | 75 |
| 200-Class IAP | 162 | 73 | 79 | 99 | 127 | 245 | 230 | 92 | 64 | 114 | 83 |

![Fig. 4. Optimized water release using the four approaches compared with the demand during the 12-month periods of 2015](image)

### 6. CONCLUSIONS

After accomplishing this study, there are some notes to be mentioned and some outputs to be remarked and highlighted.

- The key objective of this study is to invest and apply optimization algorithms and techniques on local problems. Darbandikhan dam, being one of the largest three water reservoirs in Kurdistan Region - Iraq, needs to be studied and focused on to get an optimal strategy for its operation.
  - Best results for the 100-Class model, were obtained when it is tuned on $(\alpha=3$, $\beta=1$, and $\rho=0.1)$ for IBP method and $(\alpha=3$, $\beta=1$, and $\rho=0.9)$ for IAP method.
  - For the 100-Class model the IAP method showed better performance resulting in a mean total square deviation of 0.043 units, while the IBP result was 0.105 units.
  - For both the 100 and 200 classes models, increasing the number of ants with a certain number of iterations or the opposite, results in getting better mean total square deviation values.
  - Iteration All Path (IAP) method in both 100-Class and 200-Class models of reservoir operation optimization gave better total square deviation results than Iteration Best Path (IBP) method.
  - Discretization the reservoir storage into 200 intervals resolution doesn’t give better solutions than 100 intervals due to the optimality reached in the last one
  - Large number of solutions probabilities (paths) existing in the 200 intervals makes it harder to reach the same results of the 100 intervals with the same number of iterations or ants.
  - Generally, the results from all of the four proposed models showed the high level of optimization achieved between the amount of water demand of each month and the suggested amount of water release for the same month.
  - Finally, the results of this research work show that ACO algorithm will be an active
approach for establishing optimum reservoir operation and water resources management.

7. RECOMMENDATIONS

The following recommendations are stated:

- Applying different algorithms for reservoir operation of Darbandikhan dam.
- Extending the study to include more data collected of the reservoir (minimum and maximum cases).
- Extending the study for optimizing hydro-electric power generation of the dam.

8. ACKNOWLEDGEMENT

We would like to express our gratitude to Dams and Water Resources Engineering Department specially, assistant professor Faisal A. Daham and Dr. Sarhang Mustafa who gave us the opportunity to accomplish this valuable study and supported us along the way to finish this study.

Also, many thanks to General Directorate of Dams and Reservoirs specially engineer Bery Faisal A. Daham who provided us with the necessary data about Darbandikhan Dam.

9. REFERENCES

Christian Blum (2005) ‘Ant Colony Optimization: Introduction and Recent Trends’, Physics of Life Reviews 2, 353–373, Elsevier B.V. All rights reserved Optimization, Massachusetts Institute of Technology, Artificial Life 5: 137–172

D. Nagesh Kumar and M. Janga Reddy (2005) ‘Ant Colony Optimization for Multi-Purpose Reservoir Operation’, Water Resources Management, No. 20: 879–898, ©Springer 2006

Dorian Gaertner and Keith Clark (2005) ‘On Optimal Parameters for Ant Colony Optimization algorithms’, Dept of Computing, Imperial College London, 180 Queens Gate, London, SW7 2AZ, UK

M. R. Jalali, A. Afshar, and M. A. Mariño (2006) ‘Reservoir Operation by Ant Colony Optimization Algorithms’, Iranian Journal of Science & Technology, Transaction B, Engineering, Vol. 30, No. B1, © Shiraz University

M. R. Jalali, A. Afshar, M. A. Mariño, and Hon.M.Asce (2007) ‘Multi-Reservoir Operation by Adaptive Pheromone Re-initiated Ant Colony Optimization Algorithm’, International Journal of Civil Engineering, Vol. 5, No. 4

Marco Dorigo and Christian Blum (2005) ‘Ant Colony Optimization Theory: A Survey’, Theoretical Computer Science 344, 243 – 278

Marco Dorigo, Gianni Di Caro, and Luca M. Gambardella (1999) ‘Ant Algorithms for Discrete Optimization’, Massachusetts Institute of Technology, Artificial Life 5: 137–172

R. Moeini and M.H. Afshar (2009) ‘Application of an Ant Colony Optimization Algorithm for Optimal Operation of Reservoirs: A Comparative Study of Three Proposed Formulations’, Transaction A: Civil Engineering, Vol. 16, No. 4, pp. 273-285, ©Sharif University of Technology

Rodrigo Oliveira and Daniel P. Loucks (1997) ‘Operating Rules for Multi-reservoir Systems’, Water Resources Research, Vol. 33, No. 4, Pages 839-852

Serafim Op., Lorenzo Ec., and Miodrag Is. (1991) ‘An Optimization Procedure for Water Reservoir Planning’, Yugoslav Journal of Operations Research, No. 2, Pages 225 – 233.

SMEC International Pty. Ltd., (31July 2006) Dokan and Derbendikhan Dam Inspections, World Bank.