Comparison of Genetic Algorithm and Ant Colony optimization methods for optimization of short-term drought mitigation strategies

S. MOHAMMAD MORTAZAVI N., GEORGE KUCZERA & LIJIE CUI
School of Engineering, University of Newcastle, New South Wales 2308, Australia
seyedmohammad.mortazavinaeini@studentmail.newcastle.edu.au

Abstract Drought represents a major threat to the security of water supplies. There is a variety of short and long-term options to mitigate drought impacts. In this paper, short-term strategies are explored with the goal of finding the best strategies that minimize cost. The performance of two heuristic optimization methods, Genetic Algorithm and Ant Colony, is investigated. The results show that the Ant Colony method is sensitive to the choice of method representing the decision space. It was found that Ant Colony methods were more robust and efficient than Genetic Algorithms, particularly when the number of function evaluations is limited.

Key words drought management; optimization; genetic algorithms; ant colony optimization

INTRODUCTION

As (Mays, 2007) observes, the world’s population is expanding rapidly, yet there is no more freshwater on Earth than there was 2000 years ago, when the population was less than three percent of the current six billion. Water availability is one of the most important issues facing society and will continue to be so well into the future. Therefore, it is important in urban water management to supply water in a sustainable manner in terms of quality and quantity. The challenge is particularly severe during periods of drought when sustainability of supply is threatened on a short-term basis and, if not managed properly, could affect the long-term viability of the urban area.

A variety of options is available to mitigate drought impacts. These options can be very diverse and have different effectiveness in terms of economic and social outcomes. However, only a limited number of options are practicable and potentially effective at an acceptable cost (Garrote et al., 2007). The challenge is to identify such options. Drought impact mitigation is a complex management problem, usually involving conflicting management objectives, alternative actions and different players (Traore & Fontane, 2007). The management of complex water supply systems requires advanced tools to reduce the uncertainties in hydrological and management issues, and to improve the quality of decision making for different climate conditions (Merabtene et al., 2002).

Generally actions which are useful to mitigate drought impacts can be categorized into two types, long-term and short-term. The actions taken before drought are long-term strategies. On the other hand, the actions taken after initiation of drought are referred to as short-term strategies. The most common short-term strategy in Australia is to impose restrictions on consumption.

The main aim of this paper is to identify short-term strategies which minimize cost. This is tackled by evaluating two promising optimization methods, Genetic Algorithm and Ant Colony. A case study involving the Canberra headworks system, in Australia, is used to demonstrate the capabilities of the optimization methods.

OPTIMIZATION APPROACHES

This section focuses on two optimization approaches, the well established Genetic Algorithm (GA) and the more recent Ant Colony Optimization (ACO). Both use probabilistic methods to conduct the search and thus cannot guarantee finding a global optimum. Importantly they can interface with any simulation model and therefore do not suffer from the often severe limitations of classical mathematical programming approaches.
In this particular problem, function evaluations are very expensive, especially if long stochastic flow sequences are used. For instance, (Cui & Kuczera, 2005) report a typical simulation for the Sydney headworks system taking approx 5 CPU minutes. To manage this computational problem it is important to minimize the size of the search space by carefully selecting decision variables and also to employ an efficient and robust search method. The latter motivates the focus on the efficiency and robustness of the GA and ACO methods.

Genetic algorithm

The Genetic algorithm (GA) optimization method is based on Darwin’s theory of evolution. It is a population based algorithm with every generation consisting of chromosomes which, in turn, consist of genes. There are three main operations in GAs, selection, crossover and mutation. Selection involves identifying fit parents for the crossover and mutation steps. Crossover involves a pair of parent chromosomes exchanging a portion of their bit sequence at randomly set crossover points. Mutation consists of randomly flipping each gene. Crossover and mutation operations help the GA to jump out from local optima and search the decision space more vigorously and efficiently (Cui & Kuczera, 2003).

Ant colony optimization

Ant colony optimization (ACO) is a more recent development than GA. ACO was inspired by the fact that some kinds of ants are blind yet they can find the minimum path between their nest and food. This is because of chemical substances called pheromones which ants deposit when they travel on a route. Based on this behaviour of real ants, Dorigo (1992) developed an ant colony optimization (ACO) method called Ant System (AS) to solve the travelling salesman problem (TSP) (Dorigo & Stutzle, 2004).

A variety of ACO methods have been introduced in the literature. Different methods for updating pheromones and different techniques to discipline ants to find the optimum have motivated a variety of ACO methods such as ACS, Max-Min AS, and As-ranked.

Dorigo et al. (1996) used the first ant colony optimization method called Ant System (AS) to solve the travelling salesman problem (TSP) and job-shop scheduling problem (JSP). The results showed that ACO dominated all other classical or heuristic methods and found the minimum route obtained by other methods. Dorigo & Gambardella (1997) introduced a new method based on AS called Ant Colony System (ACS). Stutzle & Hoos (1997) applied MAX-MIN AS to solve the travelling salesman problem. All of these methods have been successfully applied to a number of benchmark combinatorial optimization problems (Dorigo & Di Caro, 1999). ACO has been used in water engineering applications. For example, Zecchin et al. (2005) applied ACO to optimize water distribution systems. They undertook a parametric study for ACO parameters and compared the performance of GA and ACO for two case studies. In both case studies, ACO had a far shorter search-time than GA. However, in one case study ACO found high quality solution while in the other the solution was poor.

CANBERRA HEADWORKS SYSTEM CASE STUDY

To investigate the potential of GA and ACO to handle drought management plan optimization problems, the Canberra headworks water supply system was chosen as the case study. The Canberra system, which serves Australia’s capital city, was chosen because it had enough complexity to warrant optimization yet could be simulated on a single CPU in a reasonable time. In Fig. 1 a schematic of the Canberra system is shown. The system consists of four reservoirs referred to as Corin, Bendorra, Cotter and Googong, which are in service, and one proposed reservoir called Tennent. Also, there are several pump stations to convey water from rivers and reservoirs to Canberra.

WathNet was used to model the operation of the Canberra system. WathNet is an example of a generalized simulation model using network linear programming (NLP) to simulate the operation of a wide range of water supply headworks configurations (Cui, 2003).
Decision variables

In most drought management plans, imposing restrictions is a key element. It is usual to impose restrictions using a trigger level based on reservoir storage. Because the trigger levels must form a decreasing sequence, some care is required in specifying the decision variables. In Fig. 2 the schematic shows the sequence of trigger levels. For example, if the first trigger level is 0.6, the second trigger level must be less than 0.6. This makes the problem a constrained one.

Corin and Googong reservoirs are the two main sources of water supply for Canberra. Optimizing the balance between these reservoirs can help better to mitigate drought. To achieve this balance two decision variables, Base gain (BG) and Incremental gain (IG) are introduced in WathNet. To explain how they work, suppose there are N carryover arcs leaving a reservoir with capacity C. The penalty assigned to each arc is:

\[ \text{cost}(i) = -BG - IG(i-1), \quad i = 1, \ldots, N \]

The capacity of each arc is C/N. Because the penalty is negative the NLP tries to fill the arc to capacity. By varying BG and IG for different reservoirs, a wide range of reservoir storage target profiles can be simulated. The reservoir with higher gains (more negative carryover costs) will tend to carryover water for the next season in preference to a reservoir with lower gains.

In this study, there are six decision variables, four restriction storage trigger levels, Googong’s base gain and incremental gain.
Objective function
In the work reported here, single objective optimization was considered. The objective is to minimize total expected cost:

\[ O.F. = C_O + P_R + P_S \]  
(2)

where \( C_O \) is the expected operational cost, \( P_R \) is the penalty associated with imposing restrictions and \( P_S \) is the penalty for unplanned shortfall. Equation (3) is used to calculate restriction penalty (Cui, 2003):

\[ P_R = \frac{\varepsilon}{\varepsilon + 1} p_1 D \left[ 1 - (1 - R)^{\frac{\varepsilon}{\varepsilon + 1}} \right] \quad \text{for } \varepsilon \neq -1 \]  
(3)

where \( p_1 \) is the current price of water, \( D \) is the unrestricted demand for water, \( \varepsilon \) is the price elasticity and \( R \) is the demand reduction, assumed to be 0.95, 0.8, 0.7 and 0.65 for the four restriction levels respectively. In this study the following values were considered: \( \varepsilon = -0.25 \) and \( p_1 = $600/ML \).

The penalty for unplanned shortfalls in supply is applied to discourage the socially and politically unacceptable scenario in which the system runs out of water. Lack of water for sanitation would impact heavily on commercial activity. Moreover, many industries are dependent on water and their productivity would be reduced if supply was cut. In this study, it is assumed that the consequence of unplanned water shortage is loss of gross income which is taken as the penalty for unplanned water shortages. Noting that Canberra’s gross income for year 2006–7 was $21 billion and its water consumption was 56 148 ML, one can infer a penalty of $37 400/ML assuming loss of gross income is proportional to unplanned shortfall.

GA implementation
The GA formulation used in this case study was described by Cui & Kuczera (2003). A binary GA with tournament selection, one-point crossover and mutation in conjunction with elitism was applied to the Canberra case study.

There are two approaches for handling constraints in GAs. One is to impose a penalty on infeasible solutions and using this to augment the objective function. The other is to use a mapping to transfer the constrained search space into an unconstrained space consisting of a unit hypercube. Cui (2003) developed a mapping method to generate trigger levels using the following equation:

\[ Level(j) = Level(j + 1) + \left[ 100 - level(j + 1) \right] \times Decision(j), \quad j = N, ..., 1 \]  
(4)

where \( Level() \) denotes as storage trigger level, \( Decision() \) is the gene produced by GA and \( N \) is maximum number of trigger levels.

ACO Implementation
The ACO was implemented in the case study by first representing the search space as a graph. In the literature, two methods have been used to represent the search space. Abbaspour et al. (2001) and Kumar & Reddy (2006) used a graph similar to Fig. 3(a) and Maier et al. (2003) presented a graph similar to Fig. 3(b). In both methods, the decision variable range is split into a specific number of segments. In the first graph each variable value is represented by a node but in the second by a route. Ants travel between nodes corresponding to different variables to define a route. In the first approach called the nodal method, the route is denoted as \((k, i, j)\) which means an ant travelled from node \(i\) of variable \(k\) to node \(j\) of variable \(k+1\) while in the second one, referred to as the link method, the route is denoted as \((k,i)\) which means an ant travelled along route \(i\) from node \(k\) which corresponds to variable \(k\). To handle constraints in ACO, a taboo list is defined to prevent ants travelling on infeasible routes.

The next step is to define a transition rule which defines how ants should find their route. The transition rule for the nodal method is:
Comparison of Genetic Algorithm and Ant Colony Optimization methods

Fig. 3 (a) Schematic of ant routes travelling between five discrete variables (Nodal method) and (b) between two discrete variables (Link method).

\[
P_{kij} = \frac{[\tau_{kij}]^\alpha [\eta_{kij}]^\beta}{\sum_{j=1}^{N}[\tau_{kij}]^\alpha [\eta_{kij}]^\beta}
\]

where \( P_{kij} \) is the probability the ant at node \( i \) will travel to node \( j \); and \( \tau_{kij} \) is the pheromone trail strength and \( \eta_{kij} \) is heuristic information. The pheromone trail strength encodes a long-term memory about the entire ant search process, and is updated by the ants themselves. In contrast, the heuristic information represents a priori information about the problem or run-time information provided by a source different from the ants. The parameters \( \alpha \) and \( \beta \) are introduced to control the relative importance of the pheromone and heuristic information respectively.

Review of the literature shows there is no general rule for defining heuristic information. In this particular problem, the heuristic information for BG and IG was defined to force the ants to keep the same proportion of storage in Googong and other reservoirs. For instance, Googong’s BG is varied between 8000 and 12000 while BG for other reservoirs is 10 000. To maintain balance between Googong and the other reservoirs it is desirable to assign the same BG for all of them. In Fig. 4(a) a graph is presented to show how heuristic information is defined for BG. The maximum heuristic information is assigned to the maximum storage trigger level and the minimum heuristic information to the minimum storage trigger level.

Fig. 4 (a) Relation between heuristic information and different BG values. (b) Schematic of local search.
In this study the Max-Min AS was used. Max-Min AS differs from other ACO methods in the way it updates pheromone. In this method, the pheromone is updated just for the best iteration route or best-so-far route based on the following equations:

\[
\tau_{ij}(t+1) = (1 - \rho)\tau_{ij} + \Delta\tau_{ij}^{best}
\]  \hspace{1cm} (6)

where \( \tau_{ij} \) is the amount of pheromone on the route between node \( i \) and \( j \), \( \rho \) is a coefficient representing the evaporation of pheromone on the trail between iteration \( t \) and \( t+1 \), and \( \Delta\tau_{ij}^{best} \) is defined as:

\[
\Delta\tau_{ij}^{best} = \frac{c}{f_{best}}
\]  \hspace{1cm} (7)

where \( c \) is a positive constant and \( f_{best} \) is the optimum objective function value at iteration \( t \) or the best objective function found so far.

In the Max-Min AS method the strategy is to deposit pheromone on just the best route. Unfortunately, this strategy may result in all ants following the same route because of the excessive growth of pheromone on a good, although suboptimal, tour. To counteract this effect, lower and upper limits \( \tau_{min} \) and \( \tau_{max} \) on the possible pheromone values on any arc are imposed to avoid search stagnation.

\[
\tau_{min} \leq \tau \leq \tau_{max}
\]  \hspace{1cm} (8)

\[
\tau_{max} = \frac{1}{\rho \ast f_{best}}
\]  \hspace{1cm} (9)

\[
\tau_{min} = \frac{\tau_{max}}{z}
\]  \hspace{1cm} (10)

where \( z \) is a parameter which is assigned 150 in this study.

In this study, the objective function varies between \( 2 \times 10^6 \) and \( 300 \times 10^6 \). In this case, if the actual objective function was used to calculate \( \tau_{max} \) or \( \Delta\tau \) these parameters could be very small. For this reason the objective function is divided by \( 10^7 \) before calculating those values.

A local search can be used to improve the efficiency of Max-Min AS. When ants find the best-so-far solution, routes adjacent to that solution would be checked to find any possible improvement. In Fig. 4(b) a local search schematic is illustrated. For instance, if the third and seventh routes are assumed as the best-so-far routes, the adjacent routes, namely the second and fourth routes for IG and sixth and eighth routes for BG, are considered as a local search routes.

RESULTS AND DISCUSSION

Input data

The simulation model was run for 100 years using data generated from a stochastic model fitted to historic data. To explore the behaviour of a heavily stressed system the demand was taken as that corresponding to 175% of the current population.

Results for the variants of ACO

Because ACO and GA are probabilistic search methods, an optimization run with a different initial random number seed may produce a different optimum. This raises the question of robustness of the search. A robust method is one that shows little variation in the optimum between runs. This issue was addressed by running all methods 10 times and reporting the distribution of optimized costs.

The Min-Max AS with local search was used. Moreover, an elitist strategy in which one ant is forced to pass the best-so-far route was employed. The parameters shown in Table 1 were applied to both nodal and link implementations.
Table 1 Parameters for nodal and link ACO methods.

<table>
<thead>
<tr>
<th>Ants</th>
<th>Iteration</th>
<th>Segment</th>
<th>( \alpha )</th>
<th>( \beta )</th>
<th>( \rho )</th>
<th>( z )</th>
<th>( C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>500</td>
<td>63</td>
<td>1</td>
<td>2</td>
<td>0.1</td>
<td>150</td>
<td>0.5</td>
</tr>
</tbody>
</table>

In Fig. 5 objective function plots for the nodal and link ACO methods are presented for a single run. The link method displays superior performance with the nodal method unable to improve the objective function after about 50 iterations.

To understand this, consider Fig. 6 which presents box-plots for the objective function at each iteration. The tops and bottoms of each “box” are the 25th and 75th percentiles of the objective function values sampled by the ants, respectively. The line in the middle of each box is the data median. Outliers are displayed with a + sign. The aim of these figures is to show how ants explore the search space. In the link method the size of boxes was large over the first 50 iterations, after which the boxes shrunk but the ants continued to explore the search space. In the nodal method, however, the boxes had shrunk almost immediately with the consequence that the ants performed less vigorous exploration of the search space resulting in premature convergence.

In Fig. 7 box-plots are presented for the link method when there is no limitation on maximum pheromone. The ants converged very quickly in this case but at the expense of a greater chance of converging to a local optimum.

In Fig. 8 box-plots of the optimal cost for 10 runs for the nodal and link methods are presented. Each of the 10 runs started with a different random number seed. Figure 11 confirms the superior performance of link method. Of particular importance is that the median link cost is
very close to the lowest cost. This behaviour is attributed to the fact that the number of possible routes in the link method is less than in the nodal method and so ants can find the best route more efficiently. To illustrate this, Fig. 8(b) shows the possible routes for the Nodal and Link method when there are three segments. It shows there are nine possible routes for the Nodal method and six possible routes for the Link method. This difference becomes more significant when the number of segments is increased. Based on the superior performance of the link method, it is used in the reminder of the comparison.

To investigate the role of heuristic information, different values of $\beta$ were tested with the results shown in Fig. 9. This graph shows that use of heuristic information generally improves the ACO results. There is less variation for $\beta$ equal 1 while the best performance occurred for $\beta$ equal to 5. Moreover, for $\beta$ equal 5, the median of the costs is closest to the lowest cost.
In Fig. 10 the ACO results with and without local search are presented. It is shown that applying local search can improve ACO results. The minimum objective function achieved with local search is less than the minimum without local search. Also, with local search, the median optimized cost is closer to the best cost. However, applying local search produces greater variation between runs.

It can therefore be summarized that ACO, like other heuristic optimization methods, is sensitive to the selection of tuning parameters. To tune these parameters, sensitivity analysis must be done. Fig. 11 presents the results of a sensitivity analysis. In the last graph in Fig. 11, the product of the number of ants and iterations is fixed to give a total of 15,000 evaluations. It is clear that in this problem ACO performs better with fewer ants and more iterations. The best parameter values identified by the sensitivity analysis are summarized in Table 2.

Fig. 10 Role of local search to improve link ACO method performance.

Fig. 11 Sensitivity analysis results.
Table 2 Best parameters based on sensitivity analysis.

<table>
<thead>
<tr>
<th>Ants</th>
<th>Iteration</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\rho$</th>
<th>$z$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>750</td>
<td>1</td>
<td>2</td>
<td>0.01</td>
<td>50</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Fig. 12 Boxplots of optimized cost of ACO (tuned and untuned), two GA methods and ACO method without any limitation for $\tau_{\text{max}}$ and $\tau_{\text{min}}$ for 10 runs and 5000 evaluations.

Fig. 13 Boxplots of optimized cost of ACO (tuned and untuned), two GA methods and ACO method without any limitation for $\tau_{\text{max}}$ and $\tau_{\text{min}}$ for 10 runs and 10000 evaluations.

Fig. 14 Boxplots of optimized cost of ACO (tuned and untuned), two GA methods and ACO method without any limitation for $\tau_{\text{max}}$ and $\tau_{\text{min}}$ for 10 runs and 15000 evaluations.

Comparison of ACO and GA methods

This section compares the performance of both ACO and GA methods. In this study, a 6-bit binary GA was implemented with one point crossover. The GA parameters suggested by Cui & Kuczera (2003) were used as default values for this case study: crossover rate = 0.9; mutation rate = $1/(\text{population size})$; inversion rate = 0.7; and population size = 200. In Figs 12 to 14, five methods...
Comparison of Genetic Algorithm and Ant Colony optimization methods

are compared for 10 runs and evaluations fixed at 5000, 10 000 and 15 000 respectively. The two GA methods differed in the way the constraints on restriction triggers were handled. For the three ACO variants, two used MAX-MIN AS with untuned parameters (Table 1) and tuned parameters (Table 2), while the third had no limit on.

For the 5000 evaluation case, the untuned ACO was unambiguously the best method with the best median and minimum costs. For the other two cases, the untuned ACO still performed strongly and would probably be judged the best of the five. It is clear that not imposing limits on $\tau_{max}$ and $\tau_{min}$ resulted in the greatest variability suggesting premature convergence. However, the behaviour in Fig. 7 suggests convergence, though suboptimal, may occur rapidly. For computationally expensive simulations, it may be worth trading off loss of robustness for greater efficiency. It is interesting to note that the untuned ACO performed better than the tuned ACO. This raises questions about the worth of sensitivity analysis. Its main weakness is that it does not explicitly allow for interactions between parameters.

CONCLUSION

The main aim of this study was to investigate the performance of two heuristic methods, GA and ACO, to optimize the short-term drought mitigation strategies. The Canberra headworks system was chosen as a case study. The objective function was the total expected cost consisting of operating cost, penalty for imposing restriction and penalty for unplanned shortfall.

The results showed that the performance of ACO is sensitive to the choice of graph representing the decision tree. However, the results cast doubt on the value of using sensitivity analysis to tune ACO parameters. Overall, ACO performed more robustly than GA, especially for the case involving the fewest number of evaluations. Because drought management simulation is computationally expensive, this is a significant finding and warrants further assessment of the potential of ACO.

Acknowledgements This work was supported by funds from the eWater CRC. The authors gratefully acknowledge the assistance of Richard Barratt and Tim Purves from ACTEW who provided data and advice on the Canberra case study.

REFERENCES


