Ant Colony Optimization for Level of Service Improvements in Piping Networks

S. Christodoulou¹ and G. Ellinas²
¹ University of Cyprus, Department of Civil and Environmental Engineering, P.O.Box 20537, 1678 Cyprus. email: schristo@ucy.ac.cy
² University of Cyprus, Department of Electrical and Computer Engineering, P.O.Box 20537, 1678 Cyprus. email: gellinas@ucy.ac.cy

Abstract: The paper presents a methodology for optimal flow routing in piping networks by imitating the natural selection processes utilized by real-life ants in search of the shortest path to a food source. Ant Colony Optimization is a population-based, artificial multi-agent, general-search technique for the solution of difficult combinatorial problems with its theoretical roots based on the behaviour of real-ant colonies. The fundamental mathematical background of the ACO method is outlined and a suggested possible implementation strategy is described for solving for shortest paths in water piping networks. The ACO methodology should be of interest to both researchers and practitioners since it provides an alternative method to routing optimizations, with a wide range of applications. The described ACO virtual multi-agent approach is supplemented by a sample case study as well as algorithms for the optimization of urban water distribution networks.

Key words: Water distribution networks; ant colony optimization; pipe routing; optimization.

1. INTRODUCTION

Ongoing globalization and urbanization has brought upon urban water distribution network managers an increasing need to better understand and manage the pipe networks servicing them so as to increase the networks’ efficiency and reliability and to alleviate the strain on the usage of water resources. The increase in demand for water resources and reliable service is even more pressing in developing countries and countries with limited resources.

In the case of urban water distribution networks in areas of extended droughts and limited water resources, the problem is even more complex due to the inability of the networks’ owners to easily and cost-effectively replace utilized or lost resources, the lack of alternatives, and the pressing need to provide these resources to the public. As a result, utilities in charge of managing such water distribution networks are nowadays faced with the increasingly more complex task to intelligently and efficiently manage such networks in ways that maximize a system’s reliability and minimize its operational and management costs. In these cases, optimal pipe routing (under either normal operational conditions or emergency rerouting conditions), life-cycle costing and maintenance strategies become of paramount importance to the utilities as they seek ways to increase system reliability and quality of service while minimizing costs of operation.

Central to this balancing act of operating costs and reliability are two very important questions facing water distribution organizations: (1) what is the best way to route a network so as to minimize installation, operational and maintenance costs and increase its reliability, and (2) how can one evaluate the risk of failure for a piping network, and the proper course of action in terms of repairing or replacing a failure-prone or unreliable pipe segments? The latter of the two aforementioned problems has been addressed and reported upon extensively in literature (Andreou et. al. 1987, Aslani 2003, Vanrenterghem 2003, Christodoulou et. al. 2006, Christodoulou 2007, Christodoulou and Deligianni 2010) and several frameworks have been suggested. The work presented in this paper addresses the former of the two aforementioned problems, and to the extent
of optimizing the routes (pipe segments and valves) and operational costs of a network and for subsequently evaluating the node-to-node risk of failure and resulting impacts of failure. The goal is to seek methodologies that would allow water distribution network managers to arrive at speedier and easier evaluations of the reliability of a pipe network (or a pipe subnet) and through that at more efficient designs and management of urban piping networks. The ability to quickly assess possible pipe paths and the probability of failure from a given node to another node of interest within the system will allow the pipe network’s managers to increase the reliability-of-service within the network and to better react in the event of a catastrophic event to quickly resume operations and service-flow within the nodes of interest.

Traditional pipe routing techniques exhibit functional and computational limitations in their approach. For one, water utilities seem to route their piping networks in a “seemingly arbitrary manner” in which pipes are laid down a major road and then branched out to side streets based on consumer demand and the rate of urbanization. Secondly, the process makes no provision for longest (or shortest) paths from any node to any other node in the network. Consider, for example, the need to identify the longest and shortest paths to a desired location in the pipe network, given that at the time of evaluation the water distribution system is experiencing a failure at another location in the network. Which sequence of pipe segments and valves need to be opened or closed to redirect the flow of water to the specific location (network node) at minimum cost and time? Additionally, what is the number of customers affected by each possible rerouting scenario and how can one optimize the choice of rerouting paths? Furthermore, how can the network’s combined risk of failure be evaluated and minimized given the possible network topologies resulting from the possible pipe routings?

A possible improvement to traditional network routing practices may come in the form of intelligent path-traversing algorithms, and in particular artificial agent technologies. One such methodology is the Ant Colony Optimization (ACO) metaheuristic that imitates the collective behaviour of real-life ant colonies; a behaviour characterized by a collective knowledge-processing system that relies on knowledge acquired by individual members separately as each one “walks” a possible path. The information acquired by the individual members (trail-laying), processed and then memorized by the colony (trail-following) when in search for path solutions to a destination eventually converges to an optimal solution to the path-traversal problem; a process very similar to pipe routing and network optimization.

In computer implementations of the ACO algorithms, artificial ants are agents and solution-construction procedures that stochastically build solutions by considering (1) artificial pheromone trails which change dynamically at run time to reflect the agents’ acquired search experience, and (2) heuristic information on the problem/network being solved.

2. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a population-based general search technique, proposed by Dorigo (1992, 1996), for the solution of difficult combinatorial problems. The method is inspired by the foraging behaviour exhibited by real ant colonies and the essential characteristic of ACO algorithms is “the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of a previously obtained good solution” (Maniezo et al. 2004).

Why Ant Colony Optimization, and how can it be applied to pipe routing? The answer to this question lays in the similarities that ACO and CPM exhibit in their approach to network mapping and to longest-path or shortest-path calculations. It has been observed that real ant colonies exhibit a behaviour that is suitable to optimal network traversing and thus, by induction, to pipe routing. This behaviour is characterized by an ant colony’s ability to map a network and the possible paths within it, and to find the shortest path between a food source and an ant nest through the use of chemical aids (much similar to corporate learning and memorization). The process of network optimization
by real-rife ants is based on the ants depositing a chemical substance (termed “pheromone”) while traversing possible paths and forming “pheromone trails” which can then be followed by other ants in the colony. When choosing their way through the possible path routes, ants can sense the deposited pheromone and follow, with a stronger tendency, the paths which are marked by stronger pheromone concentrations. In essence, while an isolated ant moves essentially at random, an ant encountering a previously traversed path and pheromone-laid trail can detect such path and therefore decide with higher probability to follow the path and subsequently reinforce it with the ant’s own pheromone.

The collective behaviour is therefore characterized by a positive (reinforcing) feedback loop where the probability with which each ant chooses the path to follow increases with the number of ants having chosen the same path in the preceding steps. The final result is the relative quick convergence of the path-traversing process to the shortest path.

Since the original work by Dorigo (1991, 1992) a number of ACO algorithms has been developed, the most known of which being the Ant Colony System (ACS), the Elitist Ant System, the MAX-MIN Ant System, a rank-based version of Ant System, and the Best-Worst Ant System. The common framework for all ACO applications was proposed posteriori to be the ACO metaheuristic (Dorigo et al. 1999) while the generic problem topology and generic behavior of the artificial ants was outlined by Stützle et al. (2002). Within this framework, artificial ants are seen as stochastic solution procedures, acting as agents. These agents have short-term individual memory that helps them identify available decision options when reaching a decision node (forward pass), and additive corporate memory when the agents’ collective experience is combined to form an optimal route (backward pass). In general, the construction of a solution by the artificial ants is biased by the pheromone trails (which change at run-time), the heuristic information on the problem instance and the ants’ private memory.

ACO applications to the field of water resources management have, in recent years, increasingly been reported upon, but with very few of these applications focused on urban water distribution networks. Examples of such work are case studies on optimal management of coastal aquifers (Ataie-Ashtiani and Ketabchi 2011), continuous multi-reservoir operation optimization (Jalali et al. 2007) and large scale reservoir operation (Afshar and Moeini 2008). In the study by Ataie-Ashtiani and Ketabchi (2011) an evolutionary based approach is presented to achieve optimal management of a coastal aquifer to control saltwater intrusion. An improved Elitist Continuous Ant Colony Optimization (ECACO) algorithm is employed for optimal control variables setting of coastal aquifer management problem. The work by Jalali et al. (2007) describes a special version of a multi-colony algorithm for continuous multi-reservoir operation optimization. The proposed algorithm helps generate a non-homogeneous mesh so as to minimize the possibility of losing global optimum domain and can efficiently handle the combination of discrete and continuous decision variables. A similar application is the work by Afshar and Moeini (2008), who presented a constrained formulation of the ant colony optimization algorithm (ACOA) for the optimization of large scale reservoir operation problems.

With regards to piping networks and relevant ACO implementations, notable are the works of Haghighi et al. (2011) on the GA-ILP method for the optimization of water distribution networks, and Reca et al. (2008) on the application of several meta-heuristic techniques to the optimization of real looped water distribution networks. The former (Haghighi et al. 2011) investigates the problem of optimizing water distribution networks by means of genetic algorithms (GA) and integer-linear programming (ILP) method resulting in a hybrid optimization scheme. The latter (Reca et al. 2008) investigates the optimization of looped water distribution systems and evaluates the performance of several meta-heuristic techniques: genetic algorithms, simulated annealing, tabu search, and iterated local search. Both aforementioned works, even though not direct ACO implementations they bring forward nature-inspired methods encompassing the basics of the ACO method.
3. ANT COLONY OPTIMIZATION AND ROUTING OF PIPE NETWORKS

Since the underlying topology and path-traversing mechanism in ACO exhibits many similarities to urban water distribution systems (UWDS), the concepts and methodology employed by the ACO metaheuristic can find a direct application in pipe routing optimization. If one substitutes the search for shortest path (ACO) to the search for shortest or longest path (UWDS) and treats ACO ants, states, connections and cost function to UWDS’s water flow, operational state, pipes/valves and customers serviced respectively then the ACO metaheuristic can be employed in solving for the shortest (or longest) path in connected, acyclic graphs (such as pipe networks). Furthermore, by considering different nodal states and ant types then the search for shortest path can be accompanied with calculations for the shortest path from a node to any node, plus level-of-service improvements.

For a given network topology (urban water distribution system) defined by a graph \( G = (N,A) \), with \( N \) being the set of nodes (valves) and \( A \) being the set of arcs (pipes) connecting the subject nodes, the proposed ACO-based procedure for finding the shortest path(s) between a set of chosen nodes \( N_1 \) and \( N_2 \) can be summarized by the following steps:

1. Initialize all arcs with small amounts of pheromone, \( t_0 \). This value can be an inverse line-distance between the nodes \( N_1 \) and \( N_2 \), or the inverse line-distance of the subject arc.

2. An artificial ant is launched from node \( N_1 \) (the start node) pseudo-randomly walking from a node to a successor node via the connecting arcs until it reaches either the end-node (\( N_2 \)) or a dead end. When at a given node, the artificial ant’s selection of an arc to follow is probabilistic, based on a stochastic assignment of each \( i^{th} \) arc’s likelihood of selection, as defined by
   \[
   p_i = \frac{\tau_i \eta_i^\beta}{\sum \tau_i \eta_i^\beta} \quad (1)
   \]
   In the above equation, \( \tau_i \) is the pheromone concentration on the \( i^{th} \) arc, \( \eta_i \) is an a priori available heuristic value for the \( i^{th} \) arc and \( \beta_i \) is a parameter determining the relative influence of the heuristic information. The value of \( \eta_i \) can be defined either as the inverse of the length of the arc, or the inverse of the length of the arc plus the line-distance between the subject node and \( N_2 \). It should be noted that previously visited arcs are excluded from the selection (to enable complete “tree spanning” and avoid “memorization”). The selection is further assisted by the consideration of a randomly generated number, \( 0 \leq q \leq 1 \), which is compared to a predefined value, \( q_0 \), specific to the network topology. If \( q \leq q_0 \) then the arc with the highest value \( p_i \) is selected. Otherwise, a random selection of an arc is used based on the distribution defined by the equation for \( p_i \).

3. Upon crossing each \( i^{th} \) arc during the aforementioned solution-constructing phase a local pheromone update rule is applied to update the level of pheromone concentration at the given arc. The updated pheromone level is defined by
   \[
   \tau_i = (1 - \rho) \tau_i + \rho \tau_0 \quad (2)
   \]
   where \( \rho \) is another network topology parameter (\( 0 \leq \rho \leq 1 \)). As already noted, the goal of the local updating rule is to enable exploration of more path/route variations by making already
traversed arcs less likely to be chosen again during the randomization of the arc selection process.

4. Steps (2) and (3) are repeated for all ants in the ant colony and the most successful ant (i.e. the one whose path defines the solution) is used to globally update the network’s pheromone trails. The global update rule is defined by

\[ \tau_i = (1 - \alpha)\tau_i + \alpha \tau_L \]

where \( \alpha \) is yet another network topology parameter \( 0 \leq \alpha \leq 1 \) whose value determines the level of evaporation of pheromone concentrations. The factor \( \tau_L \) is a value inversely proportional to the path length of the best solution in case of an arc visited by the best ant or zero for all other ants. The global update rule can be applied by either the “global-best” or the “iteration-best” ant. In the first case, the ant to perform the update is the one that obtained the best solution (found the shortest path in the network) during the entire optimization process. In the second case the update is performed by the ant reaching the best solution during each iteration of the algorithm.

5. Steps (2) - (4) are repeated for either a fixed number of iterations or until a predefined condition is met, and upon termination of the algorithm the pheromone trail in the graph \( G = (N, A) \) is used to determine the solution (the arcs with highest pheromone concentration form the shortest path of the network).

It is important to note that since the desired optimization is on the shortest path of the network the calculations performed are on the negative values of the pipe lengths. This is necessitated by the fact that the aforementioned ACO algorithm maximizes the total distance between the start and end nodes. Therefore, should one desires the shortest distance then (s)he should operate on the negative values of the pipe lengths, keeping the rest of the algorithm calculations unchanged.

4. CASE STUDY NETWORK

The network topology in study (Figure 1) is a hypothetical simplified urban water distribution network that has 3 inputs (could be thought of as water reservoirs or inflow pipes), 6 main valves and 17 water distribution mains for a total length of 749 pipe-segments (each segment is actually 6 meters in length, but for the purpose of simplicity and generality the length representation will be left in generic terms).

In ACO terms, the case-study network has a topology of 10 nodes and 17 node connections and of assumed network parameters \( q_0 = 0.3, \beta = 1.0, \rho = 0.5, \alpha = 0.5, \text{ and } C = 50 \). Of the 10 network nodes included in the assumed network topology, 3 are “ant nests” (i.e. nodes with no predecessor nodes), 6 are regular nodes and 1 is a “food source” (i.e. a node with no successor nodes). It is also noted that the network in study is based on unidirectional nodal connections (an acyclic graph) so as to emulate real-life gravity-based water distribution networks, and that the network construction is based on an assumed maximum number of three successor connections per node, to simplify the manual calculations and subsequent verification by standard forward and backward-pass procedures (a Critical Path Method algorithm is used). A two-dimensional annotated view of the network topology is shown in Figure 1, also including the node-to-node lengths of all pipe segments in the network.

Let us consider the following scenario: given that a pipe breaks at the outskirts of our subnet (just before one of nodes ‘0’, ‘1’, or ‘4’), what is the best way to reroute water flow so as to guarantee service at node ‘9’? This is, in essence, a node-to-node shortest-path problem that requires evaluation of all possible paths from start-nodes ‘0’, ‘1’ and ‘4’ to end-node ‘9’. The solution path includes pipes ‘4-6’ and ‘6-9’ for a shortest-path length of 43+29= 72 units.
Let us now turn our attention to the ACO algorithm. Even though different solution states are reached at the end of each “ant run”, convergence to the correct solution is eventually achieved (in fewer than 10 iterations actually). Since the method is “intelligently” iterative (each successive iteration is stochastically dependant on previously acquired knowledge about the network topology) the shortest path may change several times before converging and stabilizing to the correct solution.

The final ACO solution results are shown in Figure 2 and tabulated in Table 1. During each iteration (“ant-run”) the algorithm generates the associated pheromone concentration levels for each arc, deduces the criticality of each connection and through that identifies the resulting shortest path (thus the critical pipe segments). The solution obtained by the ACO algorithm is tabulated in Table 1 as “FinalPheromone” values. The algorithm considers these values during the solution phase and decides which pipe segments on a continuous path are critical (column “On Shortest Path?” in Table 1).

The end-result (Table 1) is the convergence of the ACO-based process to segments ‘0-2’, ‘1-6’, ‘2-8’, ‘4-6’, ‘4-5’, ‘5-7’, ‘5-9’, ‘6-9’ and ‘7-9’ as important (high pheromone concentration). The algorithm then sifts through these pipe segments to identify the shortest continuous path from network-start to network-end and flags activities ‘4-6’ and ‘8-9’ as critical, with a calculated shortest path length of 72 units, in agreement with the solution obtained by traditional methods.
The ACO method for shortest or longest-path can be extended to account for impact calculations and for level-of-service improvements. If the pipe lengths are replaced with the number of consumers serviced by each pipe segment then the impact of each pipe burst can be linked to the number of consumers affected. Then, the ACO algorithm can be used to find the longest path in terms of customers affected and severity of event, and the shortest path to reroute the water flow so as to minimize the number of customers affected.

Table 1. Solution of the case-study network topology using ACO.

<table>
<thead>
<tr>
<th>Start Node</th>
<th>End Node</th>
<th>Arc Length</th>
<th>Pheromone Level</th>
<th>On Shortest Path</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Original</td>
<td>After 50 iterations</td>
<td>After 100 iterations</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>0.01429</td>
<td>3.69E-15</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>0.03030</td>
<td>8.88E-16</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0.05000</td>
<td>1.66E-02</td>
<td>1.43E-02</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>0.02703</td>
<td>8.88E-16</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>0.02500</td>
<td>1.15E-15</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>0.01786</td>
<td>2.96E-04</td>
<td>2.51E-03</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>0.01493</td>
<td>8.88E-16</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>0.01282</td>
<td>4.55E-16</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>0.01695</td>
<td>5.64E-03</td>
<td>4.85E-03</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.01852</td>
<td>8.88E-16</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.01852</td>
<td>9.70E-16</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.02326</td>
<td>1.39E-02</td>
<td>1.39E-02</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.03448</td>
<td>1.36E-04</td>
<td>3.37E-05</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0.02703</td>
<td>1.07E-04</td>
<td>2.64E-05</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>0.03448</td>
<td>1.31E-02</td>
<td>7.08E-03</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>0.09091</td>
<td>3.59E-04</td>
<td>8.89E-05</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0.03125</td>
<td>8.88E-16</td>
<td>0.00E+00</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

The ACO artificial agent seems to provide a powerful means to performing network optimization through shortest path calculations, since it is able to efficiently construct shortest-path solutions in acyclic (unidirectional) network topologies. Despite the seemingly iterative approach of the ACO algorithm, the method exhibits quick convergence to the final solution (as shown in the examined case study) and thus low computational time. The ACO methodology can further be modified and extended to account for reliability and level of service calculations, and node-to-node optimizations. The former can be achieved by considering the probability of failure for each pipe segment and including it in the utility function to be optimized. The latter can be achieved by setting the start node of the node-to-node sequence of interest as “ant nest” and the end node as “food source” and reconstructing the solution path (shortest path) while all other nodes are set as plain network nodes.

REFERENCES


