Application of Two Ant Colony Optimisation Algorithms to Water Distribution System Optimisation

Aaron C. Zecchin[a], Holger R. Maier[b], Angus R. Simpson[c], Michael Leonard[a], Andrew J. Roberts[d], and Matthew J. Berrisford[d]

[a] Postgraduate student, Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Adelaide.

[b] Senior Lecturer, Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Adelaide.

[c] Associate Professor, Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Adelaide.

[d] Honours student, Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Adelaide.

Abstract: Water distribution systems (WDSs) are costly infrastructure in terms of materials, construction, maintenance, and energy requirements. Much attention has been given to the application of optimisation methods to minimise the costs associated with such infrastructure. Historically, traditional optimisation techniques have been used, such as linear and non-linear programming, but within the past decade the focus has shifted to the use of Evolutionary Algorithms (EAs), for example Genetic Algorithms, Simulated Annealing and more recently Ant Colony Optimisation (ACO). ACO, as an optimisation process, is based on the analogy of the foraging behaviour of a colony of searching ants, and their ability to determine the shortest route between their nest and a food source. Many different formulations of ACO algorithms exist that are aimed at providing advancements on the original and most basic formulation, Ant System (AS). These advancements differ in their utilisation of information learned about a search-space to manage two conflicting aspects of an algorithm's searching behaviour. These aspects are termed 'exploration' and 'exploitation'. Exploration is an algorithm's ability to search broadly through the problem's search space and exploitation is an algorithm's ability to search locally around good solutions that have been found previously. One such advanced ACO algorithm, which is used within this paper, is the Max-Min Ant System (MMAS). This algorithm encourages local searching around the best solution found in each iteration, while implementing methods that slow convergence and facilitate exploration. In this paper, the performance of MMAS is compared to that of AS for two commonly used WDS case studies, the New York Tunnels Problem and the Hanoi Problem. The sophistication of MMAS is shown to be effective as it outperforms AS and performs better than any other EAs in the literature for both case studies considered.

Keywords: *Optimisation; Evolutionary Algorithms; Ant Colony Optimisation; Water Distribution Systems*

1 INTRODUCTION

Due to the high costs associated with the construction of water distribution systems (WDSs) much research over the last 25 years has been dedicated to the development of techniques to minimise the capital costs associated with such infrastructure. This process has been given the title of "optimisation" or "optimal design" of WDSs.

Within the last decade, many researchers have shifted the focus of WDS optimisation from traditional optimisation techniques based on linear and non-linear programming (e.g.[1]-[3]) to the implementation of Evolutionary Algorithms (EAs) namely; Genetic Algorithms (GAs) [4]-[8], Simulated Annealing [9], the Shuffled Frog-Leaping Algorithm (SFLA) [10], and Ant Colony Optimisation (ACO) [11], [12]. Noted advantages that exist with the use of EAs for application to WDSs are; (i) only discrete, commercial sized pipe diameters are considered, (ii) they deal only with objective function information and avoid complications associated with determining derivatives or other auxiliary information, (iii) they are global optimisation procedures (i.e. they consider points throughout the entire solution space as opposed to descent algorithms that search only locally), and (iv) as they deal with a population of solutions, numerous optimal or near-optimal solutions can be determined.

Due to the iterative nature of the solution generation of EAs, they can be intuitively seen as algorithms that incrementally search through the solution-space using knowledge gained from solutions that have already been found to further guide the search. The searching behaviour of EAs can be characterised by two main features [13], (i) *exploration*, which is the ability of the algorithm to search broadly through the solution-space and (ii) *exploitation*, which is the ability of the algorithm to search more thoroughly in the local neighbourhood where good solutions have previously been found. By definition, these attributes are in conflict with one another.

ACO is an EA based on the foraging behaviour of ants [14]. It has seen a wide and successful application to many different optimisation problems (see [15] for an overview) and recently it has been seen to perform very competitively for WDS optimisation [11]. Many different ACO algorithms have been developed, providing advancements on the initial and most simple formulation of ACO, Ant System (AS) [14]. These advancements improve the operation of ACO's decision policy (i.e. solution component selection process) and the manner in which the policy incorporates new information, to help in exploring the search space. These developments have primarily been aimed at managing the trade-off between the two conflicting search attributes of exploration and exploitation. Many notable advances on the simple AS have been developed [15], however, only one of these is considered in this paper; the Max-Min Ant System (MMAS) [16] (note, comparison is also made with the results of the ACO algorithm used in [11]).

The objective of this paper is to assess the efficacy of the additional mechanisms incorporated in the Max-Min Ant System, compared to the more basic Ant System, for WDS optimisation. To undertake this assessment, a comparison between the performance of AS and MMAS for two case studies is presented. These algorithms are also compared to the best performing algorithms previously presented in the literature for the two case studies considered.

2 THE WATER DISTRIBUTION SYSTEM OPTIMISATION PROBLEM

A water distribution system (WDS) is a network of components (e.g. pipes, pumps, valves, tanks, etc.) that transport water from a source (e.g. reservoir, treatment plant, tank etc.) to the consumers (e.g. domestic, commercial, and industrial users). The optimisation of WDSs is loosely defined as the selection of the lowest cost combination of appropriate component sizes and component settings such that the criteria of demands and other design constraints are satisfied. In practise, the design of WDSs can take many forms, as WDSs are comprised of many different components and have many different design criteria. For example, treating the design process as an optimisation problem, the decision variables within the problem could involve the selection of diameter sizes for all pipes, the sizing of tanks, selection of valve pressure settings and valve locations, pump types and pump locations. In addition to these potential decision variables, the demands on the system could involve a range of cases including peak hour, fire and extended period simulation loadings. The constraints on the system may be specified to include minimum and maximum allowable pressures at each demand point, a maximum velocity constraint for each of the pipes and water quality requirements. In addition to this, for the system to be properly assessed, a more rigorous set of design criteria could be required that quantifies the inherent uncertainty that exists within the system (examples of uncertainty include variations in nodal demands, projected growth of nodal demands and variations in the performance of components).

However, the literature on optimisation of WDSs has traditionally dealt with a much more simplistic and idealised problem. The decision variables have primarily been associated with the pipes within the system, where more specifically, the decision options have been the selection of (i) a diameter for a new pipe, (ii) a diameter for a duplicate pipe, and (iii) the cleaning of an existing pipe to reduce the hydraulic resistance. The only constraints on the system have been that minimum allowable pressures at each of the nodes are satisfied. This form of the optimisation of WDSs is used within this paper. A semi-formal expression of the optimisation problem is given in this paper, which expands on previous formulations [6], [9], as multiple demand patterns and pipe rehabilitation options are included (similar to [8]), such that the formulation encompasses problems such as the Gessler Network [4].

Within the framework outlined above, a design Ω, is defined as a set of n decisions where n is the number of pipes to be sized and or rehabilitated, that is $\Omega = (\omega_1,...,\omega_n)$ where ω_i is the selected option for pipe i, and $\omega_i \in$ ($option_{i,j}$: j = 1,… , NO_i) where $option_{i,j}$ is the j^{th} option for pipe i and NO_i is the number of options available for pipe i. For each option there is an associated cost $c_{i,j}$ of implementing that option and an action on the pipe (i.e. the placement of a duplicate pipe of a certain diameter or the cleaning of the existing pipe). The optimisation problem (i.e. the minimisation of the WDS design cost) can be expressed in the following way

$$\min \quad C(\Omega)=\sum_{i=1}^{n} L_i c(\omega_i) \tag{1}$$

Subject to

$$H_i^j(\Omega) \geq \overline{H_i^j} \quad \forall i=1,..,N_{node} \quad \forall j=1,..,N_{pattern} \tag{2}$$

$$DM_i^j + \sum_{k\in\Theta_{out}^k(\Omega)} Q_k^j(\Omega) - \sum_{k\in\Theta_{in}^k(\Omega)} Q_k^j(\Omega) = 0 \quad \forall i=1,..,N_{node}, \quad \forall j=1,..,N_{pattern} \tag{3}$$

$$H_{i_{start}}^j(\Omega) - H_{i_{end}}^j(\Omega) = A\frac{L_i}{HW_i(\Omega)^a D_i(\Omega)^b} Q_i^j(\Omega)^a \quad \forall i=1,..,N_{pipe}, \quad \forall j=1,..,N_{pattern} \tag{4}$$

where (1 is the objective and $C(\Omega)$ is the cost of design Ω, L_i is the length of pipe i, $c(\omega_i)$ is the unit length cost of ω_i; (2 is the design constraint where $H_i^j(\Omega)$ is the actual head at node i for demand pattern j and design Ω, $\overline{H_i^j}$ is the minimum allowable head at node i for demand pattern j, N_{node} is the total number of nodes and $N_{pattern}$ is the number of demand patterns.

In addition to the design constraints, the fundamental equations for fluid flow within a closed conduit must be satisfied for the set of $H_i^j(\Omega)$ to be a real solution to the hydraulic equations. These are the nodal continuity and head loss equations given in Equations 3 and 4, respectively (note, as the head loss equation is expressed in terms of the nodal head difference, the conservation of energy constraint is inherently included). DM_i^j is the demand for node i and demand pattern j, $Q_k^j(\Omega)$ is the flow in pipe k for design Ω for demand pattern j, $\Theta_{in}^i(\Omega)$ is the set of all pipes that provide flow into node i for design Ω and $\Theta_{out}^i(\Omega)$ is the set of pipes that provides flow out of node i for design Ω. The headloss equation ((4) used within this study is the Hazen-Williams equation (the formulation easily allows the Darcy-Weisbach equations to be used if desired) where $H_{i_{start}}^j(\Omega)$ is the head at the starting node of pipe i for design Ω and demand pattern j, $H_{i_{end}}^j(\Omega)$ is the head at the ending node of pipe i for design Ω and demand pattern j, $D_i(\Omega)$ is the selected diameter of pipe i for design Ω, $HW_i(\Omega)$ is the associated Hazen-Williams coefficient of the diameter selected for design Ω and pipe i, N_{pipe} is the total number of pipes including new pipes, A is a constant that is dependent on the units used and a and b are regression coefficients The adopted values of A, a and b vary throughout the literature (see [6] for an overview). The values used in this paper are consistent with those used within the hydraulic solver package EPANET2 and are A = 10.670 (for SI units), a = 1.852, and b = 4.871.

In practice, only the design constraints (i.e. (2) need to be considered, as a hydraulic solver is generally used to determine the set of $H_i^j(\Omega)$ (and corresponding $Q_k^j(\Omega)$), which automatically satisfies the continuity and headloss constraints (i.e. Equations 3 and 4). For ease of reference, the optimisation problem outlined above will be referred to as the Water Distribution System Problem (WDSP).

3 ANT COLONY OPTIMISATION

3.1 General Overview

3.1.1 Analogical origin of ACO

ACO [14], is a discrete combinatorial optimisation algorithm based upon the foraging behaviour of ants. Over a period of time a colony of ants is able to determine the shortest path from its nest to a food source. The exhibited ‘swarm intelligence’ of the ant colony is achieved via an indirect form of communication that involves the individual ants following and depositing a chemical substance, called pheromone, on the paths they travel. Over time, shorter (or more desirable) paths are reinforced with greater amounts of pheromone, as they require less time to be traversed, thus becoming the dominant paths for the colony (i.e. ants tend to follow paths that have greater amounts of pheromone on them). This operation is best explained using an example.

Consider the situation in Figure 1(a) where an ant colony has presently determined the shortest path from its nest (N) to a food source (F). Consider then interrupting this system via obstacle A-B, as shown in Figure 1(b). The ants cannot continue to follow the old pheromone trails and are required to turn left or right around the obstacle. As route-B is the shortest path, the ants that select this path will reconstitute the interrupted pheromone trail the quickest and arrive at F or N, depending on their direction of travel, before the ants that selected route-A (Figure 1(c)).

Desired location for: Figure 1

Once the ants on route-B reach F or N and re-enter the circuit they will have a higher probability of reselecting route-B, as it contains more pheromone than route-A. This is because route-B has had more ants deposit pheromone on it, as the ants on route-A have not yet completed an entire tour. Similarly, once the ants on route-A reach F or N and re-enter the circuit, they also will have a higher probability of selecting route-B due to the higher amount of pheromone it possesses (as indicated by the thicker lines in (Figure 1(d)) as a result of the larger number of ants that have already traversed this route.

Due to the decaying nature of pheromone, the longer path will eventually lose all of its pheromone as, in terms of probability, a reducing number of ants will choose to traverse it resulting in it receiving fewer and fewer deposits of pheromone. The result of this combined impact of pheromone addition and pheromone decay is that the shorter path eventually becomes the dominant path that the majority of the ants select (Figure 1(d)). Despite the simplicity of this two-decision example, the gradual convergence of the colony to the optimal solution, via the positive reinforcement of good solutions with pheromone deposits, is sufficiently illustrated.

3.1.2 ACO as an optimisation process

As with most EAs, the ACO algorithm iteratively generates populations of solutions that are a stochastic function of information that has been learned from previous iterations. To apply ACO to a combinatorial optimisation problem, it is important to outline the problem structure that ACO deals with and the nature of its solution generation. As given in [17], ACO represents a combinatorial optimisation problem by a graph $\mathcal{G}$ and a constraint set[1] Θ. The graph is defined as $\mathbb{G} = \mathbb{G}(\mathbb{N}, \mathbb{L})$ where $\mathbb{N}$ is the set of nodes and $\mathbb{L}$ is the set of edges linking the nodes. A solution to the problem is a permissible tour through $\mathbb{G}(\mathbb{N}, \mathbb{L})$, that is, a solution can be viewed as a set of edges $S \in \mathbb{S}$ where $\mathbb{S}$ is the set of solutions that satisfies the constraints Θ. A tour is constructed by an ant (i.e. decision agent) starting at some node (typically randomly selected) and incrementally selecting edges to follow based on the set of edges that are available to the ant given its semi-constructed tour. The set of permissible edges is specified by the constraint set $\Theta = \Theta(i, S')$ where $\Theta(i, S')$ is the set of edges available for selection to an ant that is at decision point i, given that the ant has the semi-constructed tour S' (i.e. the ant has followed the path S' from its initial decision point to get to decision point i). This process is continued until the ant completes its tour, that is, until the tour is an element of $\mathbb{S}$ (e.g. clearly $\Theta(i, S') = \emptyset$ for any i where $S' \in \mathbb{S}$).

For example, if the small problem depicted in Figure 1 is considered to be the problem of finding the shortest path from the nest to the food source and back to the nest again, this problem can be represented by the graph given in Figure 2 (note that this is one of many possible formulations of the graph). Within this graph there are two nodes $\mathbb{N} = \{N, F\}$ and four directed edges $\mathbb{L} = \{NF_{A,} NF_B, FN_A, FN_B\}$, where the direction is indicated by the ordering of the nodes' initials in the edge symbol and the subscript refers to the route taken. As the problem is considered to be the determination of the shortest path from N to F and back again, the constraint set can be defined by $\Theta(N, \emptyset) = \Theta(N) = \{NF_{A,} NF_B\}$ ($S' = \emptyset$ for decisions from N as it will always be the starting node) and $\Theta(F,\{NF_A\}) = \Theta(F,\{NF_B\}) = \Theta(F) = \{FN_A, FN_B\}$ (i.e. decisions from F are independent of the path chosen to get to F). Therefore the problem solution space is given by $\mathcal{S} = \{S : S = \{s_N, s_F\}, s_N \in \Theta(N), s_F \in \Theta(F)\}$, and clearly the optimal solution is $S^* = \{NF_B, FN_B\}$.

Desired location for: Figure 2

An ant's selection process from the edges in $\Theta(i, S')$ is based on a probabilistic decision policy. This policy considers a trade-off between the pheromone intensity on a particular edge and the desirability of that edge with respect to its individual influence on the objective function. The desirability has different definitions for different problems. For example, if the objective is to minimise cost, the desirability of an edge may be set equal to the inverse of the cost associated with that edge (e.g. cheaper edges are more desirable) or in the example given above, as shorter edges are more desirable, the desirability would be set equal to the inverse of the edge's length. Taking these two properties of an edge into account, ACO algorithms effectively utilise heuristic information that has been learnt (represented by pheromone intensity) in addition to incorporating a bias towards edges that are of a greater desirability. The decision policy is given by the probability function [14]

$$p_{i,j}(t) = \frac{\left[\tau_{i,j}(t)\right]^{\alpha}\left[\eta_{i,j}\right]^{\beta}}{\sum_{(i,l)\in\Theta(i,S')}\left[\tau_{i,l}(t)\right]^{\alpha}\left[\eta_{i,l}\right]^{\beta}} \quad (i,j)\in\Theta(i,S') \tag{5}$$

where $p_{i,j}(t)$ is the probability that edge (i, j) is chosen from node i in iteration t, $\tau_{i,j}(t)$ is the concentration of pheromone associated with edge (i, j) in iteration t, $\eta_{i,j}$ is the desirability of edge (i, j) and α and β are the parameters controlling the relative importance of pheromone intensity and desirability, respectively, for each ants' decision. If $\alpha >> \beta$ then the algorithm will make decisions based mainly on the learned information, as represented by the pheromone, and if $\beta >> \alpha$ the algorithm will act as a greedy heuristic selecting mainly the shortest or cheapest edges, disregarding the impact of these decisions on the final solution quality.

As with the example given in Figure 1, the evolution of the pheromone values $\tau_{i,j}(t)$ with time is at the heart of the ACO process. At the end of an iteration (i.e. after each ant has generated a solution) the pheromone value on each edge is updated. The pheromone updating rule consists of two operations; (i) a decaying operation that reduces the current level of pheromone and (ii) an additive operation, where, based on the solutions generated within an iteration, pheromone is added to an edge. The updating rule can be expressed as [14]

$$\tau_{i,j}(t+1) = \rho\tau_{i,j}(t) + \Delta\tau_{i,j}(t), \qquad \text{Equation 6}$$

where ρ is the pheromone persistence factor representing the pheromone decay (note: $0 \leq \rho \leq 1$) and $\Delta\tau_{i,j}(t)$ is the

[1] The constraint set Θ determines admissible tours through the graph $\mathbb{G}(\mathbb{N}, \mathbb{L})$ and has nothing to do with the set of constraints outlined in Equations 2 – 4 that determine the hydraulic feasibility of a solution for the WDSP. Similarly, the nodes $\mathbb{N}$ refer to graph

pheromone addition for edge (i, j). The decay of the pheromone levels enables the colony to 'forget' poor edges and increases the probability of good edges being selected (i.e. the assumption behind this is that as the process continues in time, the algorithm learns to add pheromone only to good edges, implying that more recent information is better than older information). For $\rho \to 1$, only small amounts of pheromone are decayed between iterations and the convergence rate is slower, whereas for $\rho \to 0$ more pheromone is decayed resulting in faster convergence.

The pheromone addition operation is different for each ACO algorithm and is the main feature that dictates how an ACO algorithm utilises its learned information. Typically, pheromone is only added to edges that have been selected, and the amount of pheromone added is proportional to the quality of the solution[2]. In this way, solutions of higher quality receive greater amounts of pheromone. The form of $\Delta\tau_{i,j}(t)$ for the algorithms used within this paper is discussed in more detail for each algorithm in the following sub-sections.

An example of an ACO procedure is given in Figure 3. The subroutine `initialisation_routines()` involves the initialisation of all ACO parameters, including all pheromone trails, to a specified initial value τ_0. For each iteration the process generates a solution for each ant (symbolised by the `construct_solution()` routine). During an iteration, after each ant has generated a solution, the pheromone paths are updated as given by `update_pheromone()`. Once all iterations have been looped through, the process outputs the required information and terminates.

Desired location for: Figure 3

3.2 ACO Algorithms Used Within This Study

3.2.1 Ant System

Ant System [14] is the original and most simplistic ACO algorithm. As such, it has been extremely influential in the development of more advanced ACO algorithms [15]. The decision policy and the pheromone update rule used within AS are given by Equations 5 and 6, respectively. For AS, each ant adds pheromone to all edges it has selected and consequently the pheromone addition received by each edge $(i, j) \in \mathcal{L}$ is given by [14]

nodes and not the WDS nodes of the WDSP.

[2] For minimisation problems, lower cost solutions are of a higher "quality".

$$\Delta\tau_{i,j}(t)=\sum_{k=1}^{m}\Delta\tau_{i,j}^{k}(t), \tag{7}$$

where m is the number of ants and $\Delta\tau_{i,j}^{k}(t)$ is the additional pheromone laid on edge (i, j) by the k^{th} ant at the end of iteration t. The individual pheromone addition contributed by each ant is given by [14]

$$\Delta\tau_{i,j}^{k}(t)=\begin{cases}\dfrac{Q}{f(S_k(t))} & \text{if } (i,j)\in S_k(t)\\ 0 & \text{otherwise}\end{cases}, \tag{8}$$

where Q is the pheromone addition factor (a constant), $S_k(t)$ is the set of edges selected by ant k in iteration t and $f(\cdot)$ is the objective function. From (8 it is clear that ants only add pheromone to the edges that they select and that solutions of better quality (e.g. solutions with lower $f(\cdot)$ values, as the problem is assumed to be a minimisation problem) are rewarded with greater pheromone additions.

3.2.2 Max-Min Ant System

Premature convergence to sub-optimal solutions is an issue that can be experienced by all EAs, especially those that have a greater emphasis on exploitation. To overcome this problem, the Max-Min Ant System (MMAS) was developed by [16]. The basis of MMAS is to provide *dynamically evolving bounds* on the pheromone trail intensities such that the pheromone intensity on all paths is always within a specified limit of the path with the greatest pheromone intensity. As a result all paths will always have a non-trivial probability of being selected and thus wider exploration of the search space is encouraged.

MMAS uses upper and lower bounds to ensure pheromone intensities lie within a given range, that is $\tau_{min}(t) \leq \tau_{i,j}(t) \leq \tau_{max}(t)$. The upper bound $\tau_{max}(t)$ is given by[3] [16]

$$\tau_{\max}(t)=\frac{1}{1-\rho}\frac{Q}{f\left(S^{gb}(t-1)\right)} \tag{9}$$

where S^{gb} is the global best path found up to iteration t, and the lower bound $\tau_{min}(t)$ is given by [16]

$$\tau_{\min}(t)=\frac{\tau_{\max}(t)(1-\sqrt[n]{p_{best}})}{(NO_{avg}-1)\sqrt[n]{p_{best}}} \tag{10}$$

where p_{best} is the probability that $S^{gb}(t)$ will be selected by any ant in iteration t given that all non-global best edges have a pheromone level of $\tau_{min}(t)$ and all global-best edges have a pheromone level of $\tau_{max}(t)$, n is the number of decision points and NO_{avg} is the average number of edges at each decision point. Within MMAS, the pheromone paths are initialised to an arbitrarily high value such that in the second iteration the paths are set to $\tau_{max}(t)$.

Theoretical justifications of the bounds are given in [16] but here it is sufficient to say that $\tau_{max}(t)$ is the theoretical asymptotic maximum pheromone level that an edge repeatedly receiving pheromone additions of $Q/f(S^{gb}(t))$ can achieve and $\tau_{min}(t)$ is an approximation to the pheromone level such that in the limit as $t \to \infty$, the probability that an ant selects $S^{gb}(t)$ is p_{best}. An analysis of (10 shows that lower values of p_{best} indicate tighter pheromone bounds, that is $\tau_{min}(t) \to \tau_{max}(t)$ as $p_{best} \to 0$.

As the bounds serve to encourage exploration, to provide an emphasis on exploitation, MMAS updates the iteration best ant's path, and periodically the global best path at the end of an iteration, to ensure that good information is being retained and reinforced. The updating scheme is as in Equation 6, where $\Delta\tau_{i,j}(t)$ is given by [16]

$$\Delta\tau_{i,j}(t) = \Delta\tau_{i,j}^{ib}(t) + \Delta\tau_{i,j}^{gb}(t), \tag{11}$$

where the addition from the iteration best ant $\Delta\tau_{i,j}^{ib}(t)$ is given by

$$\Delta\tau_{i,j}^{ib}(t) = \begin{cases} \dfrac{Q}{f\left(S^{ib}(t)\right)} & \text{if } (i,j) \in S^{ib}(t) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $S^{ib}(t)$ is the iteration best path found in iteration t. The pheromone addition from the global best ant $\Delta\tau_{i,j}^{gb}(t)$ is given by

$$\Delta\tau_{i,j}^{gb}(t) = \begin{cases} \dfrac{1}{f\left(S^{gb}(t)\right)} & \text{if } (i,j) \in S^{gb}(t) \text{ and } t = pf_{global} \text{ where } p \in \mathbf{N} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where f_{global} is the frequency of the global best pheromone updating and **N** is the set of natural numbers. MMAS also utilises another mechanism known as pheromone trail smoothing (PTS). This reduces the relative difference between the pheromone intensities, and further encourages exploration. The PTS mechanism is given by [16]

$$\tau_{i,j}^{*}(t) = \tau_{i,j}(t) + \delta\left(\tau_{\max}(t) - \tau_{i,j}(t)\right), \tag{14}$$

where $0 \le \delta \le 1$ is the PTS coefficient, and $\tau^{*}_{i,j}(t)$ is the pheromone intensity after the smoothing. If $\delta = 0$ the PTS mechanism has no effect, whereas if $\delta = 1$ all pheromone paths are scaled up to $\tau_{max}(t)$.

4 APPLICATION OF ANT COLONY OPTIMISATION TO WATER DISTRIBUTION SYSTEM OPTIMISATION

4.1 Transformation of constrained problem

[3][16] omit Q from their formulation, but for the sake of consistency with the adopted formulation of AS, it is included in this study.

The WDSP is a constrained optimisation problem. ACO, like all EAs, is unable to deal directly with constrained optimisation problems as, within its solution generation, it cannot adhere to constraints that separate feasible regions of a search space from infeasible regions. The standard technique to convert constrained problems to unconstrained problems is to use a penalty function. EAs direct their search solely based on information provided by the objective function. To guide the search away from the infeasible region and towards the feasible region, a penalty function increases the cost of infeasible solutions such that they are considered to be undesirable solutions. The unconstrained optimisation problem for the WDSP takes the form of minimising the sum of the real cost plus the penalty cost, that is

$$\min \quad NC(\Omega) = C(\Omega) + PC(\Omega) \tag{15}$$

where $NC(\Omega)$ is the network cost for design Ω, $C(\Omega)$ is the material and installation cost of Ω (i.e. the objective of the constrained problem) and $PC(\Omega)$ is the penalty cost incurred by Ω. Within this study, $PC(\Omega)$ was taken to be proportional to the maximum nodal pressure deficit induced by Ω as in [11]. That is

$$PC(\Omega) = \begin{cases} 0 & H_i^j(\Omega) \geq \overline{H_i^j} \ \forall i = 1,\ldots,N_{node} \ \forall j = 1,\ldots,N_{pattern} \\ \max\limits_{\substack{i=1,\ldots,N_{node} \\ j=1,\ldots,N_{pattern}}} \left\{ \overline{H_i^j} - H_i^j(\Omega) \right\} \cdot PEN & \text{otherwise} \end{cases} \tag{16}$$

where PEN is the penalty factor (constant) with units of dollars per meter of pressure violation (note, the set of heads $\{ H_i^j(\Omega) : i = 1, \ldots, N_{node}, j = 1, \ldots, N_{pattern}\}$ is calculated by a hydraulic solver). The parameter PEN is a user-defined parameter and appropriate values of PEN are different for each case study. To reduce calibrational requirements, a semi-deterministic expression for PEN derived in [12] is used, that is

$$PEN = \frac{\left[C(\Omega^{\max}) - C(\Omega^{\min})\right]}{d} \tag{17}$$

where Ω^{max} and Ω^{min} are the maximum and minimum material cost network designs, respectively, and d is a user selected pressure deficit. The value of PEN ensures that all networks with a pressure violation greater than or equal to d (an extremely small value) are made more expensive than the maximum feasible network cost.

4.2 Modification of ACO elements

As in [11], the graph $\mathbb{G}(\mathbb{N}, \mathbb{L})$ of the WDSP can be represented as a set of nodes[4] $\mathbb{N} = \{1, 2, \ldots, n + 1\}$. Each node $i \leq n$ is connected to the next via a set of directed edges $\theta_i = \{l_{i,j} : j = 1, 2, \ldots, NO_i\}$, where $l_{i,j}$ is the j^{th} edge (diameter option) connecting node i to node $i + 1$, NO_i is the number of edges connecting node i to node $i + 1$ and the set of all edges is $\mathbb{L} = \{s : s \in \bigcup_{i=1}^{n} \theta_i\}$. (To be consistent with the notation established thus far, the symbols should be $\theta_{i,k}$ to indicate the

[4] Node here does not refer to the nodes in a water distribution system network.

set of edges connecting node i to node k and $l_{i,k,j}$ to indicate the j^{th} edge connecting node i to node k, however, as node i only has edges connecting it to node i+1, the additional k subscript is unnecessary. An important point to note arising from this notational change is that for the WDSP, edge (i, j) is the j^{th} edge connecting node i to node i+1 and not the edge connecting node i to node j). A feasible tour through this graph is then an element of the solution space $\mathbb{S} = \{S : S = \{s_1, s_2, \ldots, s_n\}, s_i \in \theta_i, i = 1, \ldots, n\}$ or in reference to the terminology of section 3.1.2, Θ is independent of the semi-constructed tour and $\Theta(i, S') = \Theta(i) = \theta_i$. There are clearly many ways to formulate the graph and accompanying constraint set to describe the WDSP, however due to simplicity, and to avoid the introduction of a superficial dependence between the decision process at different decision points, this graph structure was adopted.

As the objective is to minimise cost, lower cost options are more desirable. Therefore the desirability of an option is taken as the inverse of the cost of implementing that option [11]. In other words

$$\eta_{i,j} = \frac{1}{c_{i,j}} \tag{18}$$

where $c_{i,j}$ is the unit cost of implementing diameter j at pipe i. As lower cost diameter options are more desirable, a bias in the probability towards the selection of lower cost diameters results. For options with zero cost (i.e. the null option), a virtual-zero-cost was selected.

A summary of the conversion of the general ACO problem formulation to the WDS optimisation is given in Table 1.

Desired location for: Table 1

5 CASE STUDIES

5.1 Preliminaries

Experiments were performed on two different case studies, the New York Tunnels Problem (NYTP) and the Hanoi Problem (HP). The AS and MMAS programs were coded in FORTRAN 90 with EPANET2 as the hydraulic solver. Simulations were typically performed on a dual processor 1 GHz Pentium LINUX system.

Based on a preliminary sensitivity analysis, the parameters were set to the following values. For both AS and MMAS within both of the case studies, α = 1.0, β = 0.5 and ρ = 0.98 (See [12] for a more detailed discussion of the ACO parameters). For MMAS, f_{global} was set to 10, as was the case for some of the simulations in [16]. The optimal setting for the other parameters, namely τ_0 (for AS only), p_{best} (for MMAS only) and m and Q (for both) were found to be case study dependent and were consequently calibrated independently (the guidelines determined in [12] were used for the parameters τ_0, m and Q). All results presented are based on 20 runs with different random number generator seeds.

5.2 Case Study 1: The New York Tunnels Problem

5.2.1 Background

The WDS for the NYTP is a gravity fed system from a single reservoir and consists of 20 nodes connected via 21 tunnels (Figure 4). The network details are given in Table 2. For each of the tunnels there is the option to leave the tunnel (e.g. a null option) or the option to provide a duplicate tunnel with one of fifteen different diameter sizes (Table 3).

Desired location for: Figure 4

Desired location for: Table 2

Desired location for: Table 3

As there is a null option, a virtual-zero-cost of $110 per metre was used in this study. This is approximately 1/3 of the cost of the cheapest duplicate option. This case study has a search space of approximately 1.934×10^{25} possible designs. The parameters were set as follows: τ_0 = 140 (for AS), p_{best} = 0.05, $\delta = 5\times10^{-5}$ (for MMAS), m = 90 and $Q = 2.94 \times 10^{8}$ (for both algorithms).

The known-optimum solution is $38.638 million found first by ACOA (a version of ACO with a similar updating scheme to that used by MMAS, but without the pheromone bounds) in [11] with a minimum search-time of 7,014 evaluations. It is important to note that other authors [6]-[8], [10] have proposed cheaper solutions to the NYTP, however these solutions were assessed as being infeasible by EPANET2 [11], which was the benchmark hydraulic analysis tool used in this research. In the situation where authors have presented numerous solutions, the results for the lowest cost feasible solution have been presented in this paper.

5.2.2 Results and Discussion

Table 4 shows a comparison of the two ACO algorithms with current best performing algorithms from the literature; an improved GA (GA_{imp}) [5] that uses gray coding combined with creep mutation and a variable power scaling of the fitness function, ACOA [11], and SFLA [10].

Desired location for: Table 4

From Table 4, it is seen that AS performs the worst of all the algorithms as it does not find the known-optimum, and its lowest solution deviates 1.45% from the known-optimum. MMAS was able to find the known-optimum and achieved a mean best-cost deviating only 0.51% from the known-optimum. ACOA is the only other algorithm to find the known-optimum. Even though ACOA searches more efficiently, as derived from is shorter search-times, it is known that it was not able to find the known-optimum as frequently as MMAS. MMAS is more efficient than AS and GA, but despite its better solution quality, it is less efficient than SFLA.

The improvements that the use of the pheromone bounds provide are made clear when comparing the performance of MMAS with that of ACOA. Both of these algorithms have similar updating schemes – both algorithms update the iteration-best ant's path – however, due to the pheromone bounds, MMAS is able to generally find solutions of better quality. The trade off for this is seen in MMAS' longer average search-times.

To illustrate the different behaviours of AS and MMAS, the network costs found for each evaluation number for a sample run are given in Figure 5. A total of 100 000 evaluations are shown and the function values found by AS are given in black and the values found by MMAS are given in grey. From this plot it is seen that within the first 20 000 evaluations, both algorithms were able to greatly improve the quality of solutions they found as the cost of the bulk of the solutions found tended towards the known-optimum value. It is seen that after this period of initial improvement, AS reached a point where it was unable to find solutions of increasing quality and with an increasing number of iterations the spread of the solution costs converged to a much tighter interval, implying that AS was converging.

Desired location for: Figure 5

Contrasting this performance to MMAS, it is seen that after a slightly slower rate at which the bulk of the solution qualities tended to the known-optimum value, MMAS was able to find optimal and near optimal solutions as seen by the lower values being extremely close to the known-lowest cost of $38.638 million. In time however, MMAS did not converge to the same extent as AS, but continued to generate solutions of a broad quality, indicating that, despite the fact that it located solutions of extremely high quality, the algorithm was still actively exploring the search space.

5.3 Case Study 2: The Hanoi Problem

5.3.1 Background

The Hanoi Problem (HP) has been considered by numerous authors in its discrete problem formulation [6], [8], [9]. Unlike the NYTP, it is a new design as there are no existing pipes in the system. The network consists of 34 pipes and 32 nodes organised in three loops (Figure 6). The system is gravity fed by a single reservoir and has only a single demand case. Network details are given in Table 5. For each link there are six different new pipe options where a minimum diameter constraint is enforced (i.e. no null option is available for any pipe). Table 6 gives the design options for the HP. This case study has a problem size of approximately 2.78 x 10^{26} possible designs. The parameters were set as follows: τ_0 = 26 (for AS), p_{best} = 0.5, δ = 0.0 (for MMAS), m = 80, and Q = 1.1 x 10^7 (for both algorithms).

Desired location for: Figure 6

Desired location for: Table 5

Desired location for: Table 6

The best solution given in the literature is $6.182 million found by the fast messy genetic algorithm (fmGA1) in 113 626 evaluations [8]. Again, it is important to note that other authors found solutions cheaper than this [6], [8]-[10], but these were determined as infeasible by EPANET2 (See [12] for hydraulic analysis results for these solutions). As with the NYTP, in the situation where authors have presented numerous solutions, the results for the lowest cost feasible solution have been presented in this paper.

5.3.2 Results and Discussion

Table 7 shows a comparison of the results obtained using the two ACO algorithms with those obtained using three other algorithms; GA-No.2 [6], a version of the standard GA, and fmGA1 [8]. No feasible solutions were found by AS in any run for the HP. As the lowest cost solution for the HP contains many of the larger size diameters, it can be deduced that the problem has a small feasible region, thus explaining AS’ poor performance. Other authors have also reported on the difficulty associated with this problem [10].

Desired location for: Table 7

MMAS was found to be the best performing algorithm for this case study as it was able to find a new lowest cost solution, 0.78% less than the previous lowest cost solution found by fmGA1 [8]. MMAS also achieved the lowest mean best-cost (deviating 4.24% from the new best solution) but also had the longest search-times. The relative performance of MMAS compared to that of AS is a result of its ability to explore the search space more widely for a longer period of time, resulting from its non-convergence mechanisms, but still having its search guided by only the best information, resulting from its elitist updating scheme.

A comparison of the lowest cost solution found by MMAS along with the solutions found by fmGA1 and GA-No. 2 is given in Table 8. It can be seen that the fmGA1 and GA-No.2 solutions differ from the MMAS solution by five links, which corresponds to an 85% similarity between the solutions. Of these five different links, four pipes of both the fmGA1 and GA-No.2 solutions have larger diameters than the solution found by MMAS, and only the diameter of link *[13]* is larger for the solution found by the MMAS algorithm compared with those found by the GAs. An interesting point to note is that the first 18 links of the MMAS solution are identical to those of the fmGA1 solution (except link *[13]*) and the last 15 links are identical to those of the GA-No.2 solution. These similarities correspond to the right-most loop of the MMAS solution being the same as the right-most loop of the fmGA1 solution (except for link *[19]*) and the left-most and centre loop being the same for the MMAS and GA-No.2 solutions (see Figure 6).

Desired location for: Table 8

Pressure heads for selected nodes and flows for selected links are given in Tables 9 and 10, respectively. Considering the three main links from the source feeding the network (links *[3]*, *[19]*, and *[20]*), it is seen that the MMAS solution increases the flow to the right-most arc through *[3]* and reduces the flow up the right-centre main through *[19]*. The increased flow through the right-most arc explains the need for the increase in the diameter size at link *[13]* in the MMAS solution. The extra flow in the right-most arc is used to provide greater flow to nodes along the top main (as illustrated by the larger flow rate in links *[13]* and *[14]* of the MMAS solution in Table 10) and so to reduce frictional losses, the diameter of link *[13]* was increased (see a comparison of the nodal heads at node 14, the node on the downstream end of link *[13]*, in Table 9).

Desired location for: Table 9

Desired location for: Table 10

6 CONCLUSIONS

Within this paper, the advanced ACO algorithm, MMAS, is compared to the simplistic ACO algorithm, AS, and other best performing algorithms from the literature for two WDS case studies. For both case studies MMAS is shown to outperform AS. The ease at which the water distribution system problem can be translated into the ACO problem paradigm combined with the excellent performance of MMAS illustrate that ACO is a well suited algorithm for this problem.

Within the first case study, the New York Tunnels Problem (NYTP), MMAS found the known-optimum and provided the best performance found within the literature for this case study[5] (MMAS achieved a mean objective function deviation of 0.51% from the known-optimum value). AS was unable to find the known-optimum for any runs. For the second case study, the Hanoi Problem (HP), AS performed worse than the genetic algorithm, the other ACO algorithms, and the shuffled frog leaping algorithm as it was unable to find any feasible solutions. MMAS, again provided the best performance seen in the literature[5], as it found a new lowest cost solution that was 0.78% lower than the previous lowest cost solution.

MMAS' consistently high performance for both case studies illustrates that the additional mechanisms incorporated in MMAS to manage the exploit-explore relationship are effective in improving the performance of ACO algorithms (c.f. the other ACO algorithm, AS, which performed reasonably for the NYTP but extremely poorly for the harder HP). This extremely desirable characteristic of robustness to case study type can be mostly attributed to MMAS' anti-convergence mechanisms. These were seen to enable the algorithm to search the solution space more thoroughly, whilst still being guided by the iteration-best solution. As MMAS is only one of many advanced ACO algorithms, future work should focus on the testing of the other algorithms to determine the algorithmic characteristics that are most suited to WDS optimisation.

7 REFERENCES

[1] J. Schaake, & D. Lai, Linear programming and dynamic programming applications to water distribution network

[5] Using EPANET2 as a benchmark for hydraulic feasibility.

design. *Research Report No. 116,* Department of Civil Engineering, Massachusetts Institute of Technology, (1969).

[2] P.R. Bhave, & V.V. Sonak, A critical study of the linear programming gradient method for optimal design of water supply networks, *Water Resources Research*, **28** (6) 1577-1584 (1992).

[3] K.V.K. Varma, S. Narasimhan, & S,M, Bhallamudi, Optimal design of water distribution systems using an NLP method, *Journal of Environmental Engineering*, **123** (4) 381-388 (1997).

[4] A.R. Simpson, L.J. Murphy, & G.C. Dandy, Genetic algorithms compared to other techniques for pipe optimisation, *Journal of Water Resources Planning and Management, ASCE*, **120** (4) 423-443 (1994).

[5] G.C.Dandy, A.R. Simpson, & L.J. Murphy, An improved genetic algorithm for pipe network optimization, *Water Resources Research,* **32** (2) 449–458 (1996).

[6] D.A. Savic & G.A. Walters, Genetic algorithms for least-cost design of water distribution networks, *Journal of Water Resources Planning and Management,* ASCE*,* **123** (2) 67-77 (1997).

[7] I. Lippai , P.P Heany, M. Laguna, Robust water system design with commercial intelligent search optimizers, *Journal of Computing in Civil Engrg,* ASCE*,* **13** (3) 135-143 (1999).

[8] Z.Y Wu, P.F. Boulos, C.H Orr, & J.J. Ro, Using genetic algorithms to rehabilitate distribution system, *Journal for American Water Works Association*, 74 – 85 (2001).

[9] M. Cunha, & J. Sousa, Water distribution network design optimization: Simulated Annealing Approach, *Jour. Water Resources Planning & Management,* ASCE*,* **125** (4) 215-221 (1999).

[10] M.M. Eusuff, & K.E. Lansey, Optimisation of water distribution network design using the shuffled frog leaping algorithm, *Journal of Water Resources Planning and Management*, **129** (3) 210-225 (2003).

[11] H.R. Maier, A.R. Simpson, A.C. Zecchin, W.K. Foong, K.Y. Phang, H.Y. Seah, & C.L. Tan, Ant Colony Optimization for the design of water distribution systems, *Journal of Water Resources Planning and Management,* ASCE*,* **129** (3) 200-209 (2003).

[12] A.C. Zecchin, A.R. Simpson, H.R. Maier, & J.B. Nixon, Parametric study for an ant algorithm applied to water distribution optimisation, *IEEE Transactions on Evolutionary Computation*, accepted for publication.

[13] A. Colorni, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, & M. Trubian, Heuristics from nature for hard combinatorial optimization problems, *International Transactions in Operational Research,* **3** (1) 1-21 (1996).

[14] M. Dorigo, V. Maniezzo, & A. Colorni, The ant system: optimisation by a colony of cooperating agents, *IEEE*

Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics, **26** (1) 29–41 (1996).

[15] M. Dorigo, G. Di Caro, & L.M. Gambardella, Ant algorithms for discrete optimization, *Artificial Life,* **5** (2) 137–172 (1999).

[16] T. Stützle, & H.H. Hoos, MAX-MIN Ant System, *Future Generation Computer Systems,* **16** 889–914 (2000).

[17] M. Dorigo, E. Bonabeau, & G. Therulaz, Ant algorithms and stigmergy, *Future Generation Computer Systems*, **16** 851-871 (2000).

TABLE CAPTIONS

Table 1. Conversion from the general ACO problem formulation to the WDSP

General ACO problem formulation		WDSP equivalent	
Element	**Symbol**	**Element**	**Symbol**
Path or solution. Admissible tour through the problem graph.	S	Design. Permissible set of diameter allocations to each pipe.	Ω
Edge connecting node i to node j.	(i, j)	Diameter option j available for pipe i.	$dia_{i,j}$
Set of edges available from decision point i given the semi-constructed tour S'.	$\Theta(i, S')$	Set of diameter options available for pipe i, independent of previous diameter selections.	$\theta_i = \{dia_{i,j} : j = 1, \ldots, NO_i\}$
Objective function	$f(S)$	Network cost	$NC(\Omega)$

Table 2. Network data for the New York Tunnels Problem. For brevity, data are presented in imperial units but metric units were used in the simulations with conversion factors of 1 ft = 0.3048 m and 1 in = 0.0245 m.

Link Data[a]			Node Data[b]		
Link	**Existing Diameter (in)**	**Length (ft)**	**Node**	**Demand (ft^3/s)**	**Minimum Head (ft)**
[1]	180	11600	1	Reservoir	-
[2]	180	19800	2	92.4	255
[3]	180	7300	3	92.4	255
[4]	180	8300	4	88.2	255
[5]	180	8600	5	88.2	255
[6]	180	19100	6	88.2	255
[7]	132	9600	7	88.2	255
[8]	132	12500	8	88.2	255
[9]	180	9600	9	170.0	255
[10]	204	11200	10	1.0	255
[11]	204	14500	11	170.0	255
[12]	204	12200	12	117.1	255
[13]	204	24100	13	117.1	255
[14]	204	21100	14	92.4	255
[15]	204	15500	15	92.4	255
[16]	72	26400	16	170.0	260
[17]	72	31200	17	57.5	272.8
[18]	60	24000	18	117.1	255
[19]	60	14400	19	117.1	255
[20]	60	38400	20	170.0	255
[21]	72	26400			

[a] All tunnels have a Hazen–Williams coefficient of 100

[b] Reservoir has an elevation of 300 ft and all demand nodes have zero elevation

Table 3. Design options for the New York Tunnels Problem. For brevity, data are presented in imperial units but metric units were used in the simulations with conversion factors as given in the Table 2 caption.

Option Number	Diameter (in)	Cost ($/ft)	Hazen–Williams Coefficient
1	0	0[a]	-
2	36	93.5	100
3	48	134	100
4	60	176	100
5	72	221	100
6	84	267	100
7	96	316	100
8	108	365	100
9	120	417	100
10	132	469	100
11	144	522	100
12	156	577	100
13	168	632	100
14	180	689	100
15	192	746	100
16	204	804	100

[a] A virtual unit cost of 110 $/m was used to calculate η_{i1} for all links i =1, …, 21.

Table 4. Comparison of algorithmic performance for applications to the New York Tunnels Problem. Results for AS and MMAS are based on 20 runs. NA means that the information was not available. In the situation where other authors have presented numerous solutions, the results for the lowest cost feasible solution have been presented in this table where feasibility was determined by EPANET2.

Algorithm	Best-cost ($M) *(% deviation from known-optimum)*						Search-time (evaluation number)		
	Min		mean		max		min	mean	max
AS	39.204	*(1.45)*	39.910	*(3.29)*	40.922	*(5.91)*	23,601	34,877	44,837
MMAS	38.638	*(0.00)*	38.836	*(0.51)*	39.415	*(2.01)*	22,635	30,711	42,169
GA_{imp} [a]	38.796	*(0.41)*	NA		NA		96,750	NA	NA
ACOA [b]	38.638	*(0.00)*	NA		NA		7,014	13,928	23,045
SFLA [c]	38.796	*(0.41)*	NA		NA		21,569	NA	24,817

[a] Ref. [5]

[b] Ref. [11]

[c] Ref. [10]

Table 5. Network data for the Hanoi Problem.

Link Data		**Node Data**[a, b]	
Link Number	**Length (m)**	**Node Number**	**Demand (L/s)**
[1]	100	1	Reservoir
[2]	1350	2	247.22
[3]	900	3	236.11
[4]	1150	4	36.11
[5]	1450	5	201.39
[6]	450	6	279.17
[7]	850	7	375.00
[8]	850	8	152.78
[9]	800	9	145.83
[10]	950	10	145.83
[11]	1200	11	138.89
[12]	3500	12	155.56
[13]	800	13	261.11
[14]	500	14	170.83
[15]	550	15	77.78
[16]	2730	16	86.11
[17]	1750	17	240.28
[18]	800	18	373.61
[19]	400	19	16.67
[20]	2200	20	354.17
[21]	1500	21	258.33
[22]	500	22	134.72
[23]	2650	23	290.28
[24]	1230	24	227.78
[25]	1300	25	47.22
[26]	850	26	250.00
[27]	300	27	102.78
[28]	750	28	80.56
[29]	1500	29	100.00
[30]	2000	30	100.00
[31]	1600	31	29.17
[32]	150	32	223.61
[33]	860		
[34]	950		

[a] Reservoir has an elevation of 100 m.

[b]All demand nodes have an elevation of zero and a minimum head of 30 m.

Table 6. Design options for the Hanoi Problem.

Option Number	Diameter (mm)	Cost ($/m)	Hazen–Williams Coefficient
1	304.8	45.726	130
2	406.4	70.400	130
3	508.0	98.378	130
4	609.6	129.333	130
5	762.0	180.748	130
6	1016.0	278.280	130

Table 7. Comparison of algorithmic performance for applications to the Hanoi Problem. Results for AS and MMAS are based on 20 runs. NFS means no feasible solution was found and NA means that the information was not available. In the situation where other authors have presented numerous solutions, the results for the lowest cost feasible solution have been presented in this table where feasibility was determined by EPANET2.

Algorithm	**Best-cost ($M)** ***(% deviation from optimum)***			**Search-time (evaluation number)**		
	min	**Mean**	**max**	**min**	**mean**	**max**
AS	NFS	NFS	NFS	-	-	-
MMAS[a]	6.134 *(0.00)*	6.394 *(4.24)*	6.635 *(8.17)*	35,433	85,571	113,429
GA-No. 2[b]	6.195 *(1.00)*	NA	NA	~10^6	NA	NA
fmGA1[c]	6.182 *(0.78)*	NA	NA	113,626	NA	NA

[a] A slightly better mean best-cost was found for $\delta = 0.8$, but as the difference between the means was near insignificant (i.e. less than 0.04%), $\delta = 0.05$ was considered to be the best parameter setting, as it found a significantly lower minimum best-cost solution

[b] Ref. [6]

[c] Ref. [8]

Table 8. Comparison of MMAS's minimum best-cost solution with other lowest cost solutions from the literature for the Hanoi Problem. The symbols $^{(+)}$ and $^{(-)}$ indicate an increase or reduction in diameter size, respectively, relative to the MMAS solution.

Link	Designs		
	MMAS	fmGA1	GA-No.2
[1]	40	40	40
[2]	40	40	40
[3]	40	40	40
[4]	40	40	40
[5]	40	40	40
[6]	40	40	40
[7]	40	40	40
[8]	40	40	40
[9]	40	40	40
[10]	30	30	30
[11]	24	24	30 $^{(+)}$
[12]	24	24	24
[13]	20	16 $^{(-)}$	16 $^{(-)}$
[14]	12	12	16 $^{(+)}$
[15]	12	12	12
[16]	12	12	16 $^{(+)}$
[17]	20	20	20
[18]	24	24	24
[19]	20	24 $^{(+)}$	24 $^{(+)}$
[20]	40	40	40
[21]	20	20	20
[22]	12	12	12
[23]	40	40	40
[24]	30	30	30
[25]	30	30	30
[26]	20	24 $^{(+)}$	20
[27]	12	12	12
[28]	12	12	12
[29]	16	16	16
[30]	16	16	16
[31]	12	12	12
[32]	12	16 $^{(+)}$	12
[33]	16	16	16
[34]	20	24 $^{(+)}$	20
Network Cost (M$)	6.134	6.182	6.195

[a] Ref. [8]

[b] Ref. [6]

Table 9. Pressure excess (m) for critical nodes for designs (solutions) presented in Table 8. A critical node here is defined as one where one of the three designs has a pressure excess less than 5m above the minimum allowable pressure head. Nodes with lower pressure heads than those for the MMAS solution are marked with $^{(-)}$. Hydraulic analysis was performed using EPANET2.

Node	Pressure excess (m)		
	MMAS	**fmGA1**[a]	**GA-No.2**[b]
13	0.940	1.756	4.184
14	7.247	4.275 $^{(-)}$	4.774 $^{(-)}$
15	2.952	2.072 $^{(-)}$	4.313
16	2.230	2.046 $^{(-)}$	4.313
26	2.659	3.590	3.656
27	1.705	1.998	3.100
29	1.361	2.294	1.844
30	0.419	1.846	0.973
31	0.895	1.994	1.465
32	2.166	3.496	2.764

[a] Ref. [8]

[b] Ref. [6]

Table 10. Flow rates (L/s) in main links distributing flow from source and links where the MMAS design has differing diameter sizes from the other solutions in Table 8. Links with flow less than the MMAS design are marked (–). Hydraulic analysis was performed using EPANET2.

Link	Flow rate (L/s)		
	MMAS	**fmGA1**[b]	**GA-No.2**[c]
[3][a]	2184.39	2147.45 (–)	2140.28 (–)
[11]	416.76	416.76	416.76
[13]	292.86	255.93 (–)	248.76 (–)
[14]	121.99	85.06 (–)	77.89 (–)
[16]	73.40	87.61	135.54
[19][a]	704.10	718.31	766.24
[20][a]	2167.63	2190.35	2149.59 (–)
[26]	321.39	344.11	303.36 (–)
[32]	71.32	80.80	72.52
[34]	324.15	333.63	325.35

[a] Main links leading from source (i.e. the reservoir at node 1).

[b] Ref. [8]

[c] Ref. [6]

FIGURE CAPTIONS

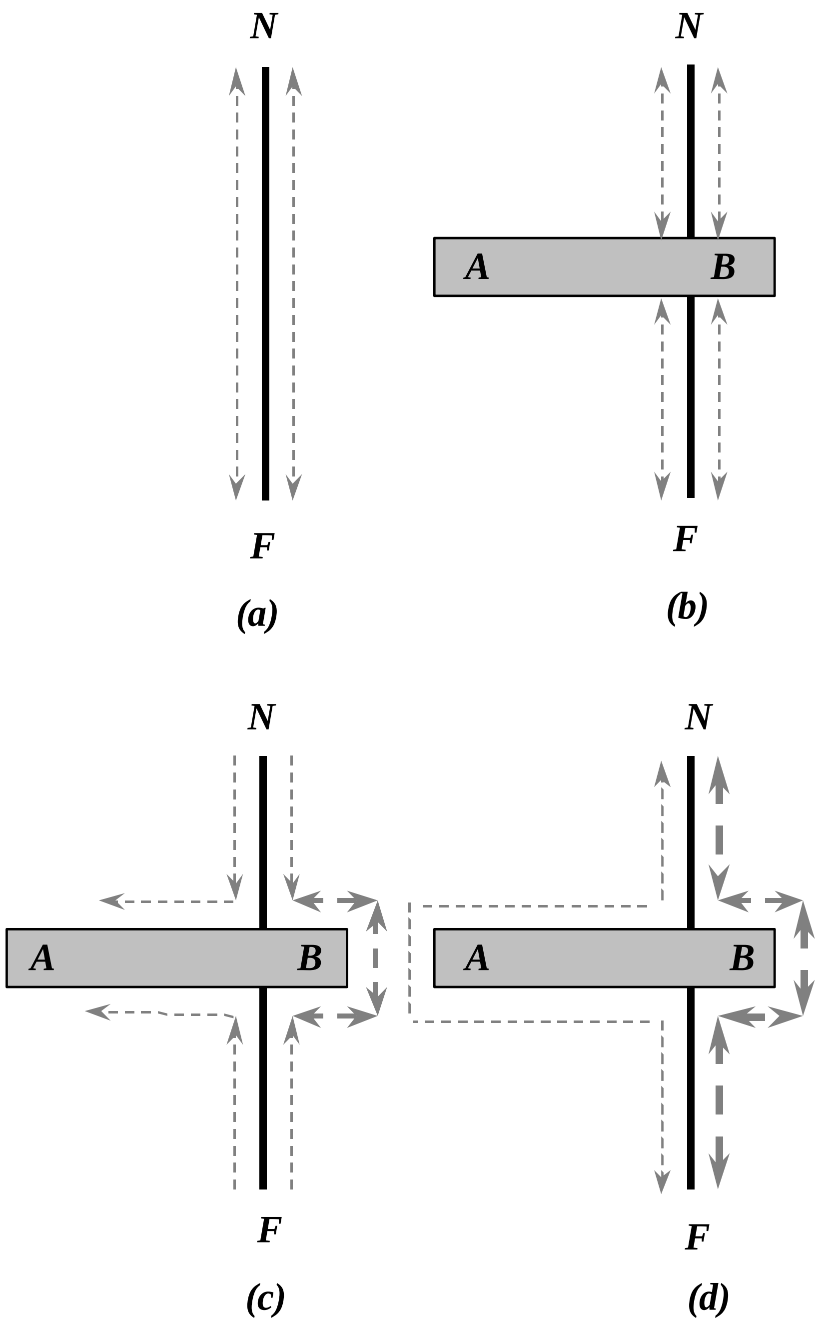

Figure 1. Illustration of the development of pheromone trails and the eventual dominance of the pheromone level of the shortest path.

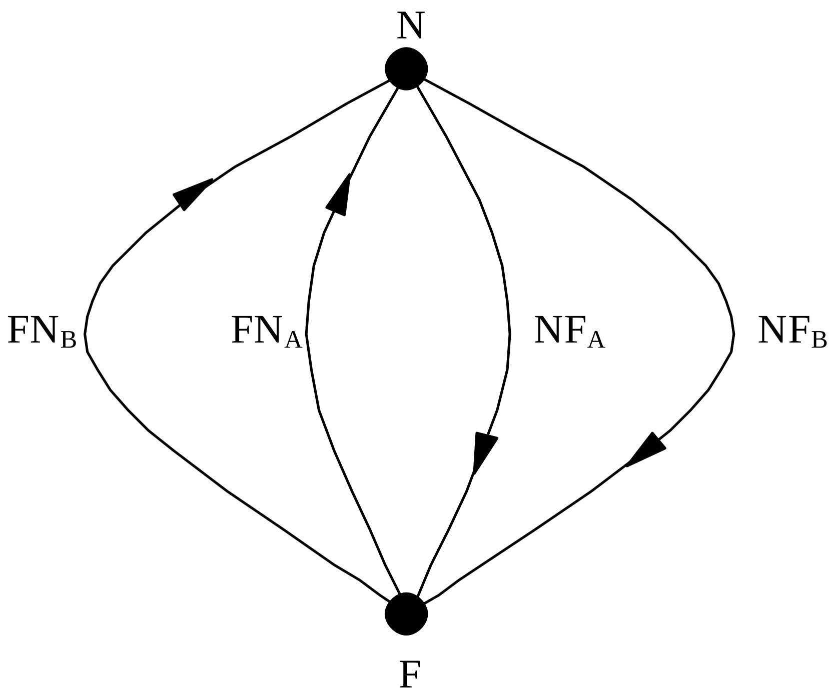

Figure 2. Possible graph representation of example given in Figure 1.

```
procedure ACO_program
  initialisation_routines()
  do (for all iterations)
    do (for all ants)
      construct_solution()
    end do
    update_pheromone()
  end do
  output_routines()
end procedure ACO_program
```

Figure 3. Example of the structure of an ACO procedure

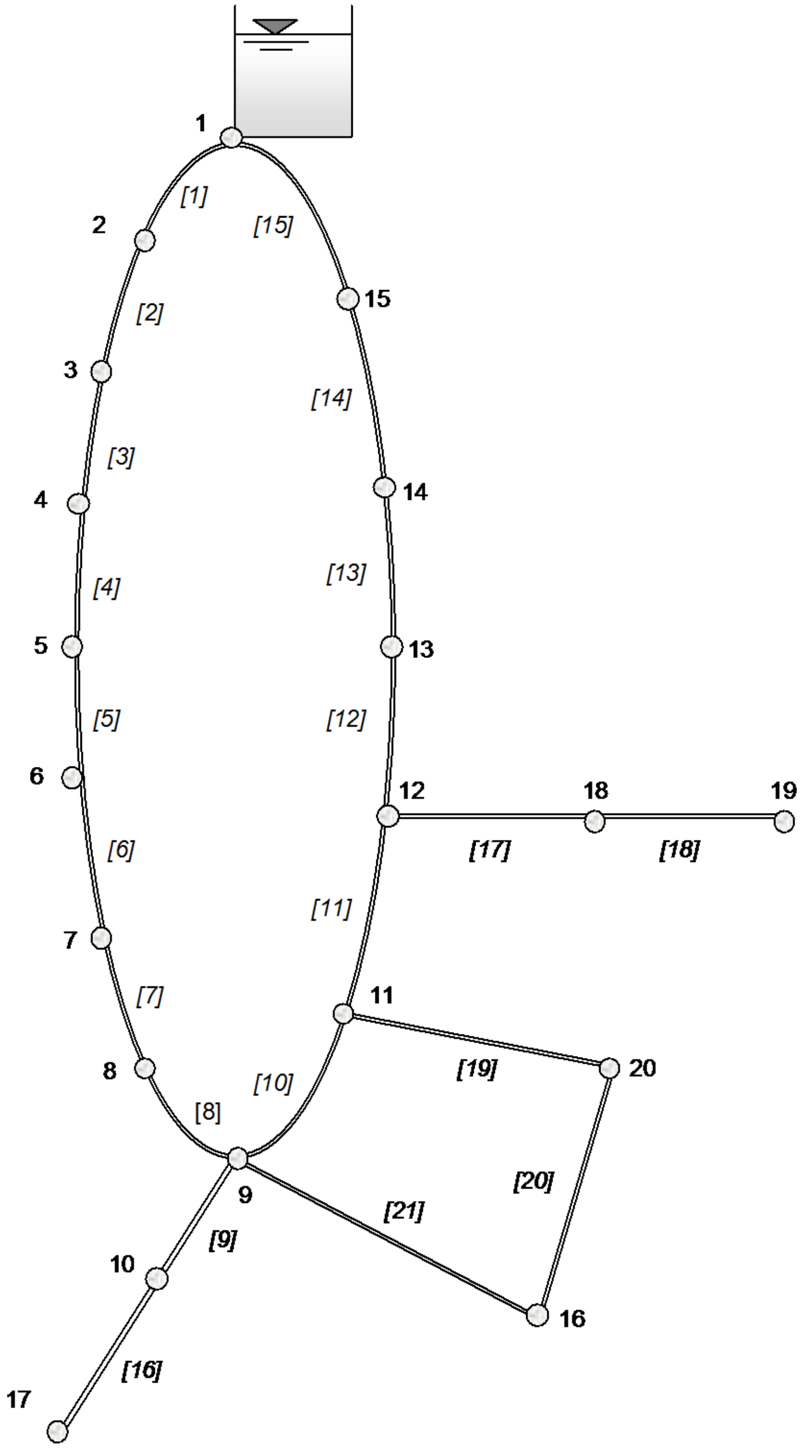

Figure 4. Network layout for the New York Tunnels Problem.

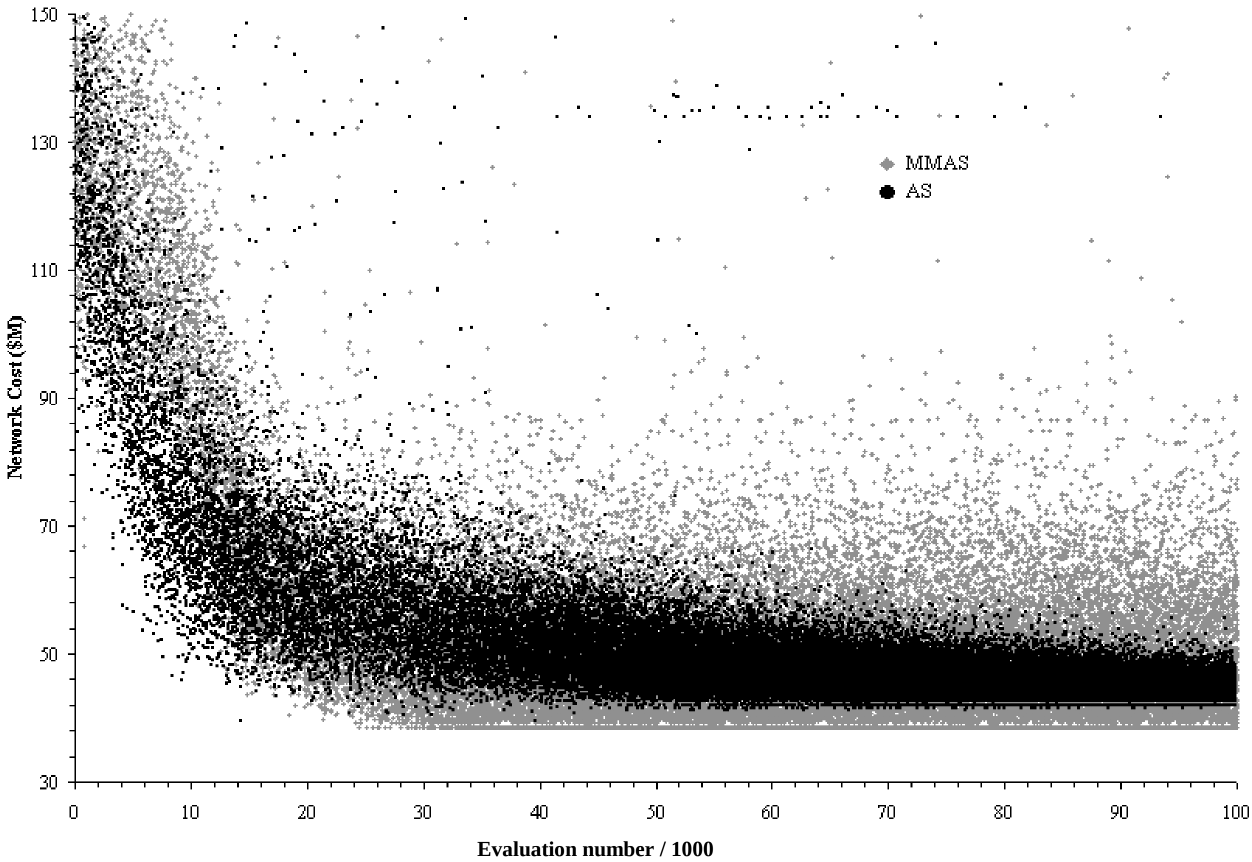

Figure 5. Network cost values found at each evaluation number (i.e. each ant within each iteration) for AS and MMAS applied to the New York Tunnels Problem.

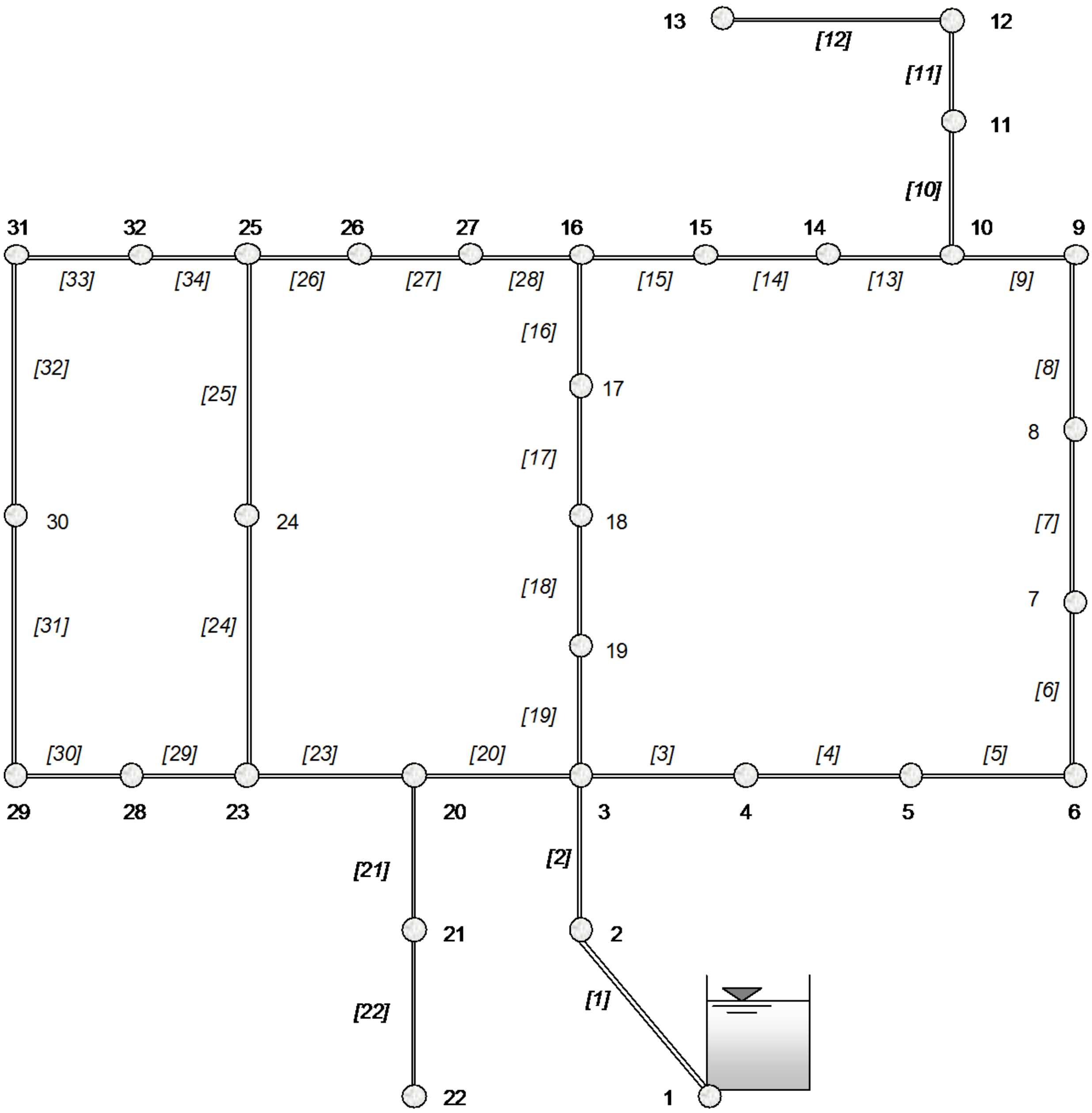

Figure 6. Network layout for the Hanoi Problem.