24

Case Study Based Convergence Behaviour Analysis of ACO Applied to Optimal Design of Water Distribution Systems

Aaron C. Zecchin, Holger R. Maier and Angus R. Simpson
School of Civil and Environmental Engineering,The University of Adelaide
Australia

1. Introduction

Based on its analogical link with nature, ant colony optimisation (ACO) aims to determine the least cost solution to an optimisation problem via the process of stigmergy (Dorigo *et al*. 2000). That is, the colony of artificial ants repeatedly stochastically constructs solutions and utilises the information gained from these solutions to guide the construction of future solutions. This process occurs in an attempt to increase the likelihood of the colony constructing the optimal solution. Each individual ant operates essentially randomly, but through alteration of its environment, a colony learns and assimilates information as a collective.

A conceptualised characteristic of this process is that the colony's searching behaviour changes with time. That is, it undergoes a highly variable, and broad reaching, initial search as the colony learns about the solution space, followed by a subsequent intensified searching in smaller regions of the solution space that the colony has learned as being promising. As such, ACO can be visualised as an initially widely spread colony converging to a point, or region, within the solution space.

Typically algorithms, such as ACO, are assessed only based on their performance in terms of the quality of the solutions found, and the computational effort required to find them. In addition to these performance based indicators, much can be learned about the different algorithms by considering the behaviour of their searching and converging process. Algorithm developers qualitatively discuss mechanisms as being exploration encouraging or exploitation encouraging (Colorni *et al*. 1996). The question arises as to the actual manifestation of these mechanisms in an algorithm's searching behaviour in terms of measurable quantities.

Within this chapter, two simple statistics for achieving this are implemented. A statistic is formulated that describes the topological nature of the *spread* of solutions through the solution space, termed the *mean colony distance*. Combining this statistic with a measure of the quality of the solutions being found, it is shown to give significant insight into the behaviour of selected ACO algorithms as the colonies converge. This chapter presents a purely computational analysis. For a theoretical treatment of ACO, the reader is referred to other work (e.g. Gutjahr, 2002).

Source: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, Book edited by: Felix T. S. Chan and Manoj Kumar Tiwari, ISBN 978-3-902613-09-7, pp. 532, December 2007, Itech Education and Publishing, Vienna, Austria

In this chapter, a case study based analysis of the convergence behaviour of four ACO algorithms applied to the water distribution system problem (WDSP) is presented. Water distribution systems are one of the largest and most costly infrastructure in all developed societies. As such, the problem of the optimal design of such systems has been a large area of active research within the Civil Engineering field since the late 1960s. The WDSP represents a difficult, highly constrained combinatorial optimisation problem.
The four ACO algorithms studied are: ant system (AS), the first and most basic form of ACO (Dorigo *et al.* 1996); the elitist AS (AS_{elite}), a version of AS utilising an elitism strategy (Dorigo *et al.* 1996); the elitist-rank AS (AS_{rank}), similar to AS_{elite} but with a rank based prioritisation of information about the solution space obtained by the ants (Bullnheimer *et al.* 1999); the max-min AS (MMAS), an ACO algorithm that bounds the evolution of the artificial pheromone trails (Stützle & Hoos 2000). On a macro level, these algorithms differ in their assimilation of new information with previously learned information. By considering the comparative convergence behaviour of these algorithms, insight into the practical outworking of their different formulations is gained.
The chapter is structured as follows. Firstly, in section 2 ACO is briefly presented and the pheromone updating mechanisms of the four algorithms are outlined. In section 3, the WDSP is explained and defined. Section 4 presents the application of ACO to the WDSP, where the issues of unconstrained problem transformation and problem graph structure are discussed. In section 5, a topology of the solution space is defined and the topological measure used to quantify the *spread* of the colony's solutions through the solution space is presented. In section 6, a detailed case study based analysis of the convergence behaviour of the algorithms is undertaken. Finally, the conclusions are given in section 7.

2. Ant Colony Optimisation Algorithms

This section is intended to provide a brief overview of ACO for the purpose of representing it in a multi-graph framework, so that its application to the WDSP in section 4 is easier to understand. For a detailed discussion of the traditional formulation, the reader is referred to Dorigo *et al.* (1999).
ACO is an evolutionary algorithmic optimisation process based on the analogy of a colony of foraging ants determining the shortest path between a food source and its nest (see Dorigo *et al.* (1996) for examples). The colony is able to *optimise* the excursions of its ants through the process of stigmergy (Dorigo *et al.* 2000), where stigmergy refers to the indirect form of communication between the ants that arises from their deposition of pheromone trails. These trails act as sign posts encouraging ants to follow them. Gradually, over time increasingly shorter pheromone trails will be reinforced with greater amounts of pheromone. This in turn will encourage more ants to follow them, potentially finding small improvements, leaving the pheromone on the less frequently used, and longer, paths to evaporate into non-existence.
ACO deals with a combinatorial optimisation problem organised as a graph $G(N, l)$, where N is the set of nodes and l is the set of edges linking the nodes (the structure of the graph is unique for each problem type). A candidate solution S to the problem is constructed by an ant selecting a feasible path through $G(N, l)$. The feasibility of the path is ensured by a special constraint function Θ, which lists the edges that are available for selection based on the previously constructed path of the ant. That is, given an ant has constructed a path S^c,

then $\Theta(S^{\epsilon})$ describes the set of edges available for selection. An ant's tour is complete when $\Theta(S^{\epsilon}) = \emptyset$, at which point, $S^{\epsilon} \in \mathbb{S}$, the set of all feasible tours through the graph.

A probabilistic decision policy is implemented at each sequential node in an ant's path for the selection of a new edge from the set Θ to add to their partially constructed path S^{ϵ}. This policy is dependent on the pheromone intensity on a particular edge (representative of the colony's learned information) and the desirability of the edge (a measure of the local effect that the selection of a particular edge will have on the value of the objective function (Dorigo *et al.* 1996)). More precisely, the probability $p_{j|S^{\epsilon}}(t)$ that edge $j \in \Theta(S^{\epsilon})$ will be selected in iteration t given an ant's partially constructed tour S^{ϵ} is

$$p_{j|S^{\epsilon}}(t) = \frac{[\tau_j(t)]^{\alpha}[\eta_j]^{\beta}}{\sum_{l \in \Theta(S^{\epsilon})}[\tau_l(t)]^{\alpha}[\eta_l]^{\beta}}, \tag{1}$$

where $\tau_j(t)$ is the pheromone concentration associated with edge j at iteration t, η_j is the desirability factor and, α and β are the parameters controlling the relative importance of pheromone and desirability, respectively, in the decision process. If $\alpha >> \beta$ 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 lowest cost options, disregarding the impact of these decisions on the final solution quality.

At the end of an iteration, all ants from the colony have constructed feasible paths through $\mathcal{G}(\mathbb{N}, \mathbb{L})$. The edge pheromone values τ_j, $j \in \mathbb{L}$ are updated to include the new information gained by the colony from the set of the new paths created by the colony $\Sigma(t) = \{S_1(t), \ldots, S_m(t)\}$, where $S_k(t) \in \mathbb{S}$ is the path chosen by ant k, and m is the number of ants in the colony. The pheromone is updated from one iteration to the next by the transitional relationship

$$\tau_j(t+1) = \rho\tau_j(t) + \Delta\tau_j(\Sigma(t), t) \tag{2}$$

where $\rho \in (0, 1)$ is the pheromone persistence factor that mimics the natural operation of pheromone decay, and governs the influence of previously learned information on future decisions, and $\Delta\tau_j(\Sigma(t),t)$ is the pheromone addition for edge j, which governs the influence of the newly acquired information from iteration t, on future decisions. The function $\Delta\tau_j(\Sigma(t),t)$ can be viewed as the value placed on edge j based on the information contained in $\Sigma(t)$, where value can be interpreted to mean the likelihood that edge j is contained in S^*, the optimal solution to the problem. Practically, this means that edge $j \in S$ is considered to have more value than $j' \in S'$ if $f(S) < f(S')$. The information in this set is essentially the resulting sample of relationships between the edges of the solutions in $\Sigma(t)$ and the corresponding function values of these solutions. The premise of ACO is that by repeated iteration of this process the colony of ants will collectively guide itself to find the optimal path through $\mathcal{G}(\mathbb{N}, \mathbb{L})$.

The main differentiating factor between ACO variants is the formulation of $\Delta\tau_j(\Sigma(t),t)$, as this describes the manner in which new information is assimilated with existing learned information. In the following subsections, the pheromone updating procedures of the four ACO variants studied in this chapter are described. All of these algorithms use the decision policy from (1).

2.1 Ant System (AS)

Ant System (AS) (Dorigo *et al.* 1996) is the original and simplest ACO algorithm. For AS, all of the ants within the colony add pheromone to their paths, and as such $\Delta\tau_j(t)$ is a function of all the solutions found at iteration t and is given by

$$\Delta\tau_j(t) = \sum_{k=1}^{m} \frac{Q}{f\left(S_k(t)\right)} I_{S_k(t)}\{j\}, \tag{3}$$

where m is the number of ants in the colony (*i.e.* the number of solutions generated at each iteration), Q is the pheromone addition factor, $f(\cdot)$ is the objective function to be minimised and $I_A\{a\}$ is the indicator function (equal to one if $a \in A$ and zero otherwise). From (3), it is clear that better solutions (*i.e.* solutions with lower objective f values) are rewarded with greater pheromone addition.

2.2 Elitist Ant System (AS_{elite})

To exploit information about the current global-best solution, Dorigo *et al.* (1996) proposed the use of an algorithm known as Elitist Ant System (AS_{elite}). This algorithm uses *elitist ants*, which only reinforce the path of the current global-best solution after every iteration (analogous to elitism strategies used in genetic algorithms). Thus, the pheromone addition is given by

$$\Delta\tau_j(t) = \sum_{k=1}^{m} \frac{Q}{f\left(S_k(t)\right)} I_{S_k(t)}\{j\} + \sigma \frac{Q}{f\left(S_{gb}(t)\right)} I_{S_{gb}(t)}\{j\} \tag{4}$$

where the first part of (4) corresponds to the pheromone addition from the colony, as defined for AS in (3), and the second part corresponds to the pheromone addition from the elitist ants, where σ is the number of elitist ants and $S_{gb}(t)$ is the set of edges comprising the global best solution found up until iteration t (*i.e.* this is equivalent to the addition of pheromone from σ ants). The updating rule for AS_{elite} allows for exploration, as each of the m solutions found by the colony receives a pheromone addition, but also encourages exploitation, as the global-best path is reinforced with the greatest amount of pheromone. As σ increases, so does the emphasis on exploitation.

2.3 Elitist-Rank Ant System (AS_{rank})

Proposed by Bullnheimer *et al.* (1999), the Elitist-Rank Ant System (AS_{rank}) further develops the idea of elitism used in AS_{elite} to involve a rank-based updating scheme. In AS_{rank}, σ elitist ants reinforce the current global-best path, as in AS_{elite}, and the ants that found the top $\sigma - 1$ solutions within the iteration add pheromone to their paths with a scaling factor related to the rank of their solution. The pheromone update formula for AS_{rank} is given by

$$\Delta\tau_j(t) = \sigma \frac{Q}{f\left(S_{gb}(t)\right)} I_{S_{gb}(t)}\{j\} + \sum_{k=1}^{\sigma-1} (\sigma - k) \frac{Q}{f\left(S_{(k)}(t)\right)} I_{S_{(k)}(t)}\{j\}, \tag{5}$$

where the first part of (5) corresponds to the addition from the elitist ants, and the second part from the ranked ants, where $S_{(k)}(t)$ is the set of edges selected by the k^{th} ranking ant in iteration t. The edges that are selected by the k^{th} ranking ant receive pheromone equivalent

to the addition from $(\sigma - k)$ ants. The potential advantages of this formulation, compared with AS and AS_{elite}, are (i) only the top $\sigma - 1$ ranked ants are used to lay pheromone (and not all m ants), which allows for the retention of only good information, and (ii) greater importance is given to the higher ranking ants (*i.e.* the top ranked solution receives $\sigma - 1$ times the normal amount of pheromone and the $(\sigma - 1)^{th}$ ranked solution receives only the normal pheromone amount), so that better edges receive more pheromone.

2.4 Max-Min Ant System (MMAS)

To overcome the problem of premature convergence whilst still allowing for exploitation, Stützle and Hoos (2000) developed the Max-Min Ant System (MMAS). The basis of MMAS is the provision of dynamically evolving bounds on the pheromone trail intensities such that the pheromone intensity on all paths is always within a specified lower bound $\tau_{\min}(t)$ of a theoretically asymptotic upper limit $\tau_{\max}(t)$, that is $\tau_{\min}(t) \le \tau_j(t) \le \tau_{\max}(t)$ for all edges j. As a result of stopping the pheromone trails from decaying to zero, all paths always have a non-trivial probability of being selected, and thus wider exploration of the search space is encouraged. The pheromone bounds at iteration t are given by (Stützle & Hoos 2000)

$$\tau_{\min}(t) = \frac{\tau_{\max}(t)\left(1 - \sqrt[n]{P_{best}}\right)}{\left(NO_{avg} - 1\right)\sqrt[n]{P_{best}}}, \quad \tau_{\max}(t) = \frac{1}{1-\rho}\frac{Q}{f\left(S^{gb}(t)\right)} \tag{6}$$

where P_{best} is the (user selected) probability that the current global-best path, $S_{gb}(t)$, will be selected, 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 $\tau_{\max}(t)$, and NO_{avg} is the average number of edges at each decision point. It should be noted that lower values of P_{best} indicate tighter bounds.

As the bounds serve to encourage exploration, MMAS adds pheromone only to the iteration-best ant's path $S_{(1)}(t)$ at the end of an iteration in order to ensure the exploitation of good solutions. To further exploit good information, the global-best solution $S_{gb}(t)$ is updated every T_{gb} iterations. The MMAS pheromone update is given by

$$\Delta\tau_j(t) = \frac{Q}{f\left(S_{(1)}(t)\right)} I_{S_{(1)}(t)}\{j\} + \frac{Q}{f\left(S_{gb}(t)\right)} I_{S_{gb}(t)}\{j\} \cdot I_{\mathbb{N}}\left\{t/T_{gb}\right\}, \tag{7}$$

where the first part of (7) corresponds to the addition from the iteration best ant, and the second part from the global best ant, where $\mathbb{N}$ 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 operation is given by

$$\tau_j(t) \leftarrow \tau_j(t) + \delta\left(\tau_{\max}(t) - \tau_j(t)\right), \tag{8}$$

where $0 \le \delta \le 1$ is the PTS coefficient. If $\delta = 0$, PTS has no effect, whereas if $\delta = 1$, all pheromone paths are scaled up to $\tau_{max}(t)$. The pheromone updating process of MMAS can be summarised as the three step process of: (i) decay and addition by (2) and (7), (ii) bounding by (6), and (iii) smoothing by (8).

3. The Water Distribution System Optimisation Problem

Water distribution systems (WDSs) consist of the system of pipes, pumps, valves *etc.* that delivers water from sources to consumers. From an optimisation perspective, the water distribution system problem (WDSP) is defined as the selection of the lowest cost combination of appropriate component sizes (*e.g.* pipes) and settings (*e.g.* valve settings) such that the criteria of water demands and other design constraints (*e.g.* minimum pressures) are satisfied. A simple example of this is as follows. Consider two networks, the first comprising pipes with small diameters and the second comprising pipes with large diameters. The first network has a low cost, but as the pipe diameters are small, the frictional pressure loss through the network will be greater, which is likely to result in insufficient pressure at the demand points (nodes). The second system is likely to provide more than adequate pressure, as the pipe diameters are large, but is also more expensive. The optimal design is the least cost combination of pipe sizes that are able to provide adequate pressure at each of the nodes. Within the WDSP, the decision variables are associated with the pipes within the system where, more specifically, the design options are the following, (i) a diameter for a new pipe, (ii) the cleaning of an existing pipe to reduce the hydraulic resistance, and (iii) no action.

As outlined in Zecchin *et al.* (2005), for the WDSP, a design involves the selection of a series of design options for all or some of the pipes within the network. A WDS design $\Omega = \{\Omega_1, ..., \Omega_n\}$ is defined as a set of n decisions where n is the number of pipes to be sized and/or rehabilitated, and Ω_i is the selected option for pipe i, and is defined as $\Omega_i \in \Lambda_i = \{\omega_{i,1}, \ldots, \omega_{i,NO_i}\}$, where Λ_i is the set of all NO_i options available for pipe i. For each option there is an associated cost, $c(\Omega_i)$, of implementing that option, and one of three actions as listed above.

The constraints on a solution are categorized as design constraints and hydraulic constraints. A design constraint is an inequality constraint that defines the minimum acceptable performance of a design, whereas hydraulic constraints are equality constraints that describe the distribution of the flow of water through the WDS (based on the fundamental equations for fluid flow within a closed conduit and the governing equations for fluid flow within a looped network). The design constraint for the WDSP specifies the minimum allowable pressure at each node, and is given as

$$H_{i,j} \geq \underline{H}_{i,j} \quad \forall i = 1,.., N_{node} \quad \forall j = 1,.., N_{pattern} \,, \tag{9}$$

where $H_{i,j}$ is the actual head at node i for demand pattern j, $\underline{H}_{i,j}$ 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.

The hydraulic equations for fluid flow within a WDS are the conservation of mass and the pipe headloss equations. As the fluid is assumed to be incompressible, the conservation of mass equations dictate that the flow rate into a node is equal to the flow rate out of a node. This can be expressed as

$$\underline{Q}_{i,j} + \sum_{k \in \Theta_{u,i}} Q_{k,j} - \sum_{k \in \Theta_{d,i}} Q_{k,j} = 0 \qquad \forall i = 1,.., N_{node} \quad \forall j = 1,.., N_{pattern} \,, \tag{10}$$

where $Q_{i,j}$ is the demand for node i and demand pattern j (by definition, a positive demand is one that draws water from the node), $Q_{k,j}$ is the flow in pipe k for demand pattern j, $\Theta_{u,j}$ is the set of all pipes for which node i is the upstream node, and $\Theta_{d,j}$ is the set of pipes for which node i is the downstream node (note that the sign convention is that positive pipe flow occurs from upstream to downstream).
The headloss equation is written as (Streeter & Wylie 1997)

$$H_{i_u,j} - H_{i_d,j} = r_{\Omega_i} \left| Q_{i,j} \right| Q_{i,j}^{\,a-1} \qquad i = 1, ..., N_{pipe},\ j = 1,.., N_{pattern}\ , \tag{11}$$

where r_{Ω_i} is a hydraulic resistance term associated with decision Ω_i, a is the flow exponent, and N_{pipe} is the number of pipes, including new pipes. The headloss equation used within most WDSPs is the Hazen-Williams equation, for which r_{Ω_i} is expressed as

$$r_{\Omega_i} = A C_{\Omega_i}^{-a} D_{\Omega_i}^{-b} L_i \qquad \forall i = 1,.., N_{pipe} \quad \forall j = 1,.., N_{pattern}\ , \tag{12}$$

where L_i is the length of pipe i, D_{Ω_i} is the diameter of pipe i for design Ω, C_{Ω_i} is the Hazen-Williams coefficient for pipe i for design Ω, 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 are those used in the hydraulic solver software EPANET2 (Rossman 2000).
The objective is the minimization of the material and installation costs, and so the WDSP can be expressed as

$$\min \ \ C(\Omega) = \sum_{i=1}^{n} c(\Omega_i), \text{ Subject to (9) – (11)} \tag{13}$$

where $C(\Omega)$ is the cost of design Ω and $c(\Omega_i)$ is the cost of decision Ω_i. As is seen from (13), despite the simplicity of the objective function, the complexity of the optimisation problem arises from the nonlinear nature of the constraints dependency on the design options Ω_i.

4. Application of Ant Colony Optimisation to Water Distribution System Optimisation

4.1 Transformation of constrained problem

The WDSP is a constrained optimisation problem. ACO 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 (here feasibility refers to constraints (9)-(11) and not the Θ function). The standard technique to convert constrained problems to unconstrained problems is to use a penalty function (Coello Coello 2002). ACO algorithms 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 poor quality 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

$$\min \quad NC(\Omega) = C(\Omega) + PC(\Omega) \tag{14}$$

where $NC(\Omega)$ is the network cost for design Ω, $C(\Omega)$ is the material and installation cost given by (13) and $PC(\Omega)$ is the penalty cost incurred by Ω. When evaluating a potential design, the set of heads $\{H_{i,j} : i = 1, \ldots, N_{node}, j = 1, \ldots, N_{pattern}\}$ is calculated by a hydraulic solver. Therefore (10)-(11) are automatically satisfied, and hence, only (9) is required to be considered in the penalty cost. Within this study, $PC(\Omega)$ was taken to be proportional to the maximum nodal pressure deficit induced by Ω as in Maier *et al.* (2003). That is

$$PC(\Omega) = \begin{cases} 0 & \text{if } H_{i,j} \geq \underline{H}_{i,j},\ i = 1,\ldots,N_{node},\ j = 1,\ldots,N_{pattern} \\ \left(\max_{(i,j)\in[1,N_{node}]\times[1,N_{pattern}]} \left\{ \underline{H}_{i,j} - H_{i,j} \right\} \right) \cdot PEN & \text{otherwise} \end{cases} \quad (15)$$

where PEN is the penalty factor (user defined) with units of dollars per meter of pressure violation.

4.2 Modification of ACO elements

As in used in previous studies (Maier *et al.* 2003; Zecchin *et al.* 2005; Zecchin *et al.* 2006; Zecchin *et al.* 2007), but formalised here, the graph $G(N, L)$ of the WDSP can be represented as a multi-graph, with the set of nodes $N = \{1, 2, \ldots, n + 1\}$. Each node $i \leq n$ is connected to the next via a set of directed edges $\theta_i = \{(i, i+1)_j : j = 1, 2, \ldots, NO_i\}$, where $(i, i+1)_j$ is the j^{th} edge 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 $L = \{s : s \in \theta_1 \cup \ldots \cup \theta_n\}$. The edge set θ_i is associated with the set of design options Λ_i, and the edge $(i, i+1)_j$ is associated with option $\omega_{i,j}$. A solution S, associated with design Ω, is a feasible tour through the graph and is an element of the solution space $S = \{S : S = \{s_1, \ldots, s_n\}, s_i \in \theta_i, i = 1, \ldots, n\}$, where the constraint function Θ is given by $\Theta(\{s_1, \ldots, s_i\}) = \theta_i$ for $i \leq n$.

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 (Maier *et al.* 2003). In other words

$$\eta_{(i,i+1)_j} = 1 / c(\omega_{i,j}) \cdot \quad (16)$$

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.

4.3 Parameter Settings

One of the limitations of ACO is that an extensive calibration phase is required to determine appropriate parameter settings. From an extensive analysis of ACO applied to the WDSP, Zecchin *et al.* (2005) determined a series of parameter guidelines relating the five fundamental ACO parameters (α, β, ρ, Q, τ_0, and m) to WDSP characteristics (such as the number of decision points n, the average number of options per decision NO_{avg}, and the cost of key design configurations such as Ω^{max}, the maximum cost design, and Ω^*, the optimum, or near optimum, design). These are summarised in Table 1.

Contrary to other problem types (Dorigo & Gambardella 1997), Zecchin *et al.* (2005) found that, for the WDSP, better performance was achieved when the ants gave greater emphasis to the learned pheromone values τ as opposed to the visibility values η, as manifested

through $\alpha > \beta$. Better performance was achieved when the pheromone persistence factor was relatively high, facilitating slow convergence and long memory times for learned information. Zecchin *et al.* (2005) showed that the ratio of Q to τ_0 is important (not the actual values of each) and empirical guidelines were determined accordingly. The best number of ants m was also found to be dependent on the number of options per decision, not just the number of decisions, as for other problem types (Dorigo *et al*. 1996).

Parameter	Heuristic
α	1.0
β	0.5
ρ	0.98
Q	$C(\Omega^{\max})$
τ_0	$Q\sqrt{nNO_{avg}}\,/\,NC(\Omega^{*})$
M	$n\sqrt{NO_{avg}}$

Table 1. Parameter guidelines for ACO parameters from Zecchin *et al.* (2005)

In addition to the guidelines derived for the ACO parameters, the following semi-deterministic expression for *PEN* was derived in Zecchin *et al.* (2005)

$$PEN = \left|C(\Omega^{\max}) - C(\Omega^{\min})\right| \Big/ \Delta H \tag{17}$$

where $\Omega^{\min}$ is the minimum cost network design, and ΔH is a user selected pressure deficit, based on the maximum acceptable pressure deficit for a feasible solution as defined by (9). The value of *PEN* ensures that all networks with a pressure violation greater than or equal to ΔH (an extremely small value) are made more expensive than the maximum feasible network cost $C(\Omega^{\min})$.

5. Analysis of Algorithm Convergence Behaviour

The standard approach to the analysis of optimisation algorithms is to assess their performance on a particular problem from statistics based on the lowest cost achieved by the algorithm (termed *best-cost*) and the computational time required for the algorithm to find the associated solution (termed *search-time*). A richer understanding of the performance of an algorithm can be achieved by considering statistics from the solutions generated by the algorithms during their run-time. A typical approach used by many authors (Simpson *et al.*, 1994; Cunha & Ribeiro, 2004; Afshar & Marino, 2007) is to track the minimum cost generated in each iteration as a means of assessing the algorithm's convergence behaviour. This statistic is important, as it indicates the effectiveness of the search, but acts only as a surrogate measure of the actual convergence behaviour of the algorithm.

This work aims to extend this qualitative performance assessment to include a topologically based statistic to describe an algorithm's convergence behaviour. From the perspective of ACO, convergence is defined as the point in time at which all ants select the same path

through the problem graph (*i.e.* the colony's population of solutions is fixed at a certain point in the solution space $\mathcal{S}$) from that point onward. Thus, convergence behaviour is the nature of the colony's solution generation up until the point of convergence. Topologically, convergence means that the distance between all solutions generated by the colony is zero. Conversely, a non-converged search will have some *spread* of the solutions throughout the solution space. It is the quantification and tracking of this *spread* that is of interest in describing an algorithm's convergence behaviour.

The motive behind convergence analysis is to gain a greater understanding of how the different explorative and exploitative mechanisms in the ACO algorithms considered actually impact the algorithm's search. Below, the topology of the solution space is first defined, and then the adopted convergence metric, the *mean colony distance*, is presented.

It is important to note that the use of metrics is widely used in evolutionary algorithm based multi-objective optimisation (Deb 2001). However, this is fundamentally different to what is considered here. In multi-objective optimisation, the distribution of solutions throughout the multi-dimensional objective space is of primary interest, and thus the metrics operate in this space. Conversely, this chapter is concerned with the distribution of solutions within the solution space, and, as such, the mean colony distance is defined on this space.

5.1 Topology of the Solution Space

Fundamental to any topologically based statistic is the notion of distance between points (solutions) in the solution space. A measure of distance for all elements within the set $\mathcal{S}$ is equivalent to defining a metric $d : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_+$ associated with $\mathcal{S}$ that defines the distance between two elements S, $S' \in \mathcal{S}$ (Cohen 2003). For sets whose elements have no specific numerical relation, the Hamming distance is a natural metric. This has been used by Bose *et al.* (1994) and Stützle & Hoos (2000) for the travelling salesperson problem. A generalisation that applies to sets whose elements are equal length lists of objects is

$$d(S, S') = \sum_{i=1}^{n} d_i(s_i, s_i'), \tag{18}$$

where $S=\{s_1, \ldots, s_n\}$, $S'=\{s_1', \ldots, s_n'\}$, $s_i, s_i' \in \theta_i$ and $d_i : \theta_i \times \theta_i \rightarrow \mathbb{R}_+$ is itself a metric for the set of all possible i^{th} elements in the list. For the Hamming distance, $d_i(\cdot,\cdot)$ is either zero or one, depending whether s_i and s_i' are equal or not. However, if the elements in the set have some other attribute that can be exploited, such as a meaningful ordering based on some property, then the metric can be defined so as to include this information.

Considering (12), it is seen that the selection of an option Ω_i is essentially equivalent to selecting a resistance parameter r_{Ω_i}. Therefore, it is meaningful to say that an option is closer to one option than another based purely on the relative differences between their associated resistance parameter values. The list of options Λ_i for pipe i can therefore be meaningfully ordered by the magnitude of their associated resistance parameter. That is, consider the following ordering of Λ_i, based on the resistance parameter $\underline{\Lambda}_i = \{\underline{\omega}_{i,1}, \ldots, \underline{\omega}_{i,NO_i}\}$, where $r_{\underline{\omega}_{i,1}} \leq \ldots \leq r_{\underline{\omega}_{i,NO_i}}$, and the distance d_i between any two of these options $\underline{\omega}_{i,j}$ and $\underline{\omega}_{i,k}$, is given by

$$d_i(\underline{\omega}_{i,j}, \underline{\omega}_{i,k}) = |j - k| \quad \text{where } \underline{\omega}_{i,j}, \underline{\omega}_{i,k} \in \Lambda_i \,. \tag{19}$$

In this context, the distance between two options is the number of positions in the ordered list $\underline{\Lambda}_i$ that separates the two options.

5.2 Mean colony distance

By ascribing a topology to the search space, the colony of solutions generated within an iteration can be considered to be spread, in some manner, over the topology. This spread of solutions gives an indication of how widely, or tightly, an algorithm is searching. To use the terminology of Colorni *et al.* (1996), whether the algorithm is *exploring* broadly through the search space or *exploiting* smaller regions of the search space. In order to quantify this spread, the mean of the distances between each of the ants' solutions has been used in this chapter, which is henceforth referred to as the mean colony distance d_Σ. Mathematically this is given as the summation of the distances of each unique pair of solutions divided by the total number of pairs, and is expressed as the map $d_\Sigma : \mathbb{S}^m \rightarrow \mathbb{R}_+$ where

$$d_\Sigma(t) = \frac{2}{m(m-1)} \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} d\left(S_k(t), S_l(t)\right), \tag{20}$$

where $m(m-1)/2$ is the number of unique pairs that exist in a colony of m ants. The usefulness of d_Σ as a behavioural analysis measure is fully realised when considering its variation with time. For example: periods of high exploration when solutions are spread broadly throughout the search space correspond to large values of d_Σ; periods during which the algorithm converges correspond to a series of decreasing d_Σ values; the point at which the algorithm converges is given by $d_\Sigma = 0$, as this indicates that all solutions in $\Sigma(t)$ are equal. As such, d_Σ provides a direct measure of an algorithm's convergence behaviour.

6. Case Studies

Experiments were performed on four different case studies, the Two Reservoir Problem (TRP), the New York Tunnels Problem (NYTP), the Hanoi Problem (HP) and the Doubled New York Tunnels Problem (2-NYTP). The ACO algorithms were coded in FORTRAN 90 with EPANET2 (Rossman 2000) as the hydraulic solver. Parameter settings from Zecchin *et al.* (2005), summarised in Table 1, were used for parameters α, β, ρ, τ_0, m, and Q for all algorithms with the adjustment that τ_0 was scaled by σ for AS_{elite} and AS_{rank} (in accordance with the logic of the derivation of Q in Zecchin *et al.* (2005)) and for MMAS, τ_0 was set to an arbitrarily high number, as proposed by Stützle & Hoos (2000). For AS_{elite} and AS_{rank}, σ required calibration for each case study. For MMAS, f_{global} was set to 10, as in Stützle & Hoos (2000) and P_{best} and δ were calibrated for each case study. The best-cost and search-time statistics for AS, AS_{elite}, and AS_{rank} and MMAS are as presented in Zecchin *et al.* (2007).

6.1 Case Study 1: Two-Reservoirs Problem

6.1.1 Preliminaries

The TRP was initially studied by Gessler (1985), and Simpson *et al.* (1994) introduced the metric version. The TRP is a 14-pipe network with two reservoirs (Figure 1). The TRP involves three demand cases: a peak hour demand case and two fire loading demand cases. There are nine existing pipes, of which three are considered for rehabilitation, duplication with one of eight pipe sizes, or to be left alone. There are five new pipes that must be sized

with one of eight diameters. The reader is referred to Simpson *et al.* (1994) for case study details. The problem, consists of 32,768,000 possible combinations.

6.1.2 Results

Based on the heuristics given in Table 1, $\{\tau_0, m\} = \{27, 25\}$ and preliminary testing showed that a maximum number of iterations of $I_{max} = 400$ was sufficient for the algorithms to not significantly improve on their solution quality after this point. For each algorithm, a single run for the TRP consisted of 10,000 function evaluations. The range of parameters tested was: $\sigma \in [2, 20]$ for AS_{elite}; $\sigma \in [2, 20]$ for AS_{rank}; $\{P_{best}, \delta\} \in [1 \times 10^{-5}, 0.99] \times [0, 0.99]$ for MMAS. AS_{elite} achieved a mean performance within 1% of the known optimum for most of the tested values of σ, with better performances observed using $3 \leq \sigma \leq 5$. Similarly, AS_{rank} achieved a mean performance within 1% of the known optimum for all tested values of $\sigma > 2$ with lower mean best-cost values occurring for $10 \leq \sigma \leq 14$. AS_{rank} tended to be less sensitive to variations in σ than AS_{elite}, as it was able to find the optimum in each run for a broader range of values for this parameter. MMAS achieved a mean performance within 1% of the optimum for values of $P_{best} \geq 0.001$ and $\delta \leq 0.001$, with the solution quality deteriorating for parameter values outside these ranges. The optimal parameter values were as follows: for AS_{elite}, $\sigma = 4$; for AS_{rank}, $\sigma = 10$; for MMAS, $\{P_{best}, \delta\} = \{0.5, 10^{-6}\}$.

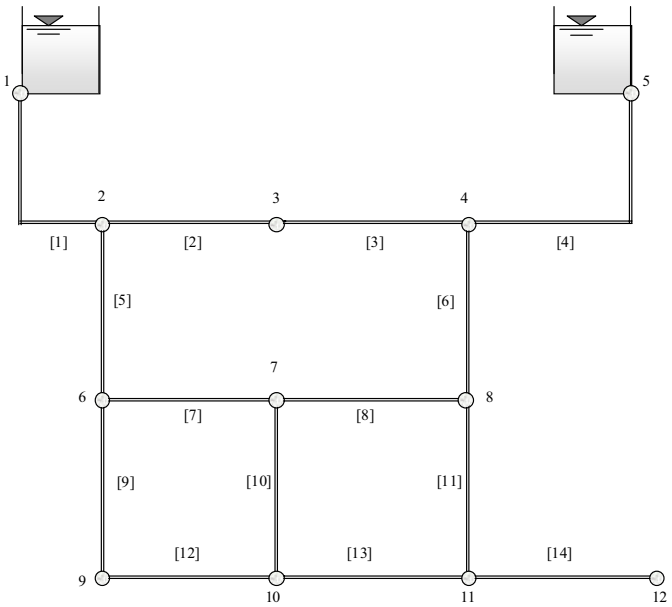

Figure 1. Network layout for the Two-Reservoir Problem

Table 2 gives a comparison of the results obtained using the ACO algorithms considered and those obtained from a selection of other best performing algorithms that have been applied to the discrete version of the TRP previously. A detailed statistical analysis of these algorithms was given in Zecchin *et al.* (2007), but it is clear that all algorithms performed extremely well (finding the optimum for all 20 runs) and were, comparatively, computationally efficient.

Plots of the iteration best-costs $f_{min}(t)$ and the mean colony distance $d_{\Sigma}(t)$, averaged over 20 runs, are given in Figure 2 (a) and (b). In addition to this, other run-time properties (to be discussed) are given in Figure 2 (c). With regard to $f_{min}(t)$, three distinct phases are observed. The first part of the search, phase-I, is a relatively short phase in which all algorithms find relatively poor quality solutions, which is followed by the second phase, phase-II, in which a dramatic increase in solution quality (reduction in the minimum cost) takes place, which leads into the third phase, phase-III, in which the rate of increase of the solution quality plateaus and the algorithms seem to not find any better solutions (or in some cases, the optimum is found repeatedly).

Algorithm	**Best-cost ($M)** *(% deviation from optimum)*						**Mean search-time (evaluation no.)**
	Minimum		**Mean**		**Maximum**		
AS	1.750	*(0.00)*	1.750	*(0.00)*	1.750	*(0.00)*	2,084
AS_{elite}	1.750	*(0.00)*	1.750	*(0.00)*	1.750	*(0.00)*	1,842
AS_{rank}	1.750	*(0.00)*	1.750	*(0.00)*	1.750	*(0.00)*	1,523
MMAS	1.750	*(0.00)*	1.750	*(0.00)*	1.750	*(0.00)*	2,993
PE[a]	1.834	*(4.80)*	-		-		900
GA_{prop}[b]	1.750	*(0.00)*	1.759	*(0.51)*	1.812	*(3.54)*	23,625
GA_{tout}[c]	1.750	*(0.00)*	1.750	*(0.00)*	1.750	*(0.00)*	8,700
SA[d]	1.750	*(0.00)*	NA		NA		NA
ACOA[e]	1.750	*(0.00)*	1.769	*(1.09)*	1.813	*(3.60)*	12,455
TS[f]	1.728[i]	-	NA		NA		~10,000
$ACOA_{i\text{-}best}$[g]	1.750	*(0.00)*	1.750	*(0.00)*	1.750	*(0.00)*	8,509
ACS[h]	1.750	*(0.00)*	1.770	*(1.13)*	1.904	*(8.81)*	5,014

[a] Partial enumeration (Gessler 1985). [b] GA based on a proportionate selection rule (Simpson *et al.* 1994). [c] Tournament selection GA (Simpson & Goldberg 1994). [d] Simulated Annealing (Sousa & Cunha 1999). [e] An AS variant that subtracts pheromone (Maier *et al.* 2003). [f] Tabu Search (Cunha & Ribeiro 2004). [g] Iteration-best updating version of ACOA (Maier *et al.* 2003). [h] Ant Colony System (Zecchin *et. al* 2007). [i] Not feasible by complete enumeration results (Simpson *et al.* 1994).

Table 2. Comparison of performance of AS, AS_{elite}, AS_{rank}, MMAS, and other algorithms from the literature applied to the Two-Reservoir Problem. Results for AS, AS_{elite}, AS_{rank}, and MMAS are based on 20 runs. NA means that the information is not available

These three phases can also be seen clearly when considering the behaviour of d_{Σ} in Figure 2 (b). To make the distinction between the phases clearer, the bar chart in Figure 2 (c) indicates when the algorithms are in each of the phases (dark grey for phase-I, light grey for phase-II and the remaining white space for phase-III). For d_{Σ}, phase-I corresponds to a brief period of extremely broad searching where almost no convergence behaviour is displayed, followed by

phase-II, in which relatively rapid convergence is observed, and phase-III, in which the rate of convergence either plateaus or decreases gradually to $d_\Sigma(t) = 0$, the point of convergence.

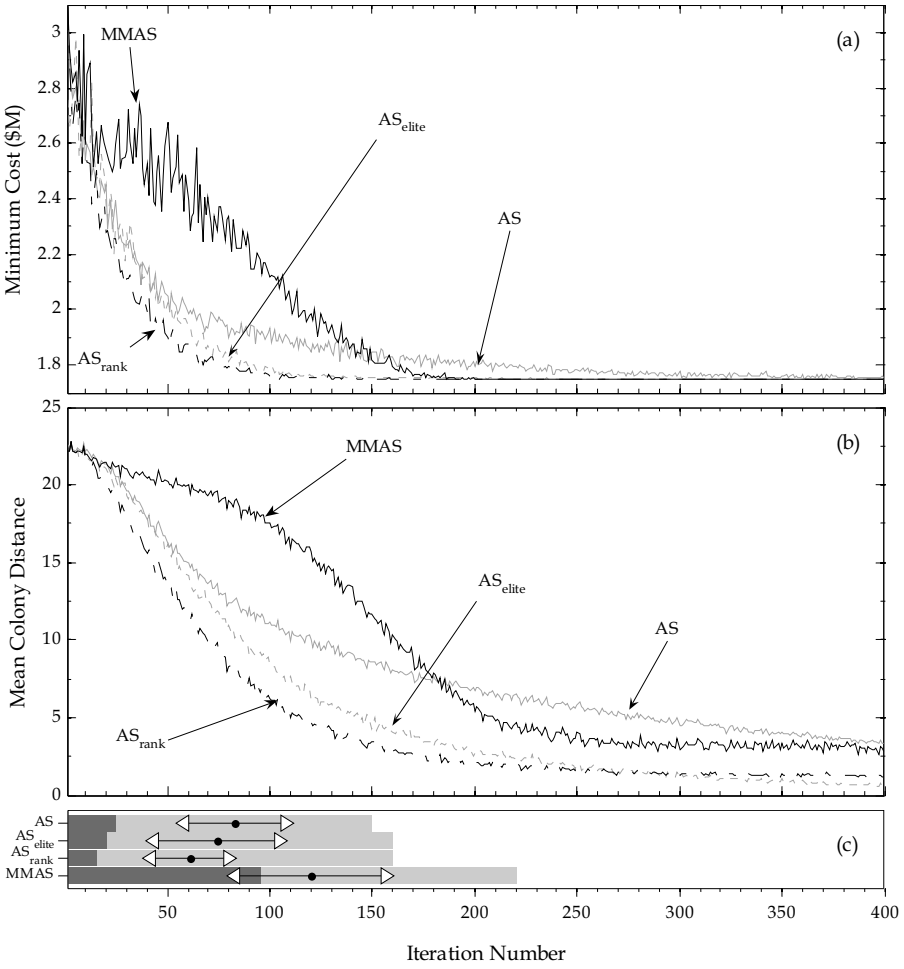

Figure 2. Plots of (a) the minimum cost ($M) found in each iteration $f_{min}(t)$, (b) the mean colony distance $d_\Sigma(t)$, and (c) run-time statistics for AS, AS_{elite}, AS_{rank}, and MMAS applied to the Two Reservoir Problem. Plots (a) and (b) are averaged from 20 runs. Plot (c) depicts the three convergence phases: phase-I (dark grey); phase-II (light grey); phase-III (remaining white space). The line graphs overlaying the bar charts in (c) indicate the search-time statistics (based on 20 runs) with the dot indicating the mean search-time, and the left and right arrows indicating the mean minus and plus a standard deviation, respectively

The nature and time spent in each of these three phases is different for each algorithm. As seen in Figure 2, AS, AS_{elite}, and AS_{rank} have a relatively short broad searching phase-I, followed by a rapid convergence in phase-II. In contrast, MMAS has a relatively long broad

searching phase-I, followed eventually by rapid phase-II convergence. The relatively long phase-I for MMAS may be attributed to the exploration encouraging mechanisms of pheromone bounding and pheromone smoothing. AS_{elite} and AS_{rank} have faster phase-II convergence than AS, which can possibly be attributed to the elitist exploitation mechanisms in these algorithms driving the search to converge faster. In phase-III, AS and AS_{elite} experience a gradually reducing, but steady, convergence, albeit AS_{elite} in a much tighter region after phase-II. In contrast to this, AS_{rank} and MMAS plateau in their convergence, as seen by $d_\Sigma(t)$ tending to a constant value in phase-III.

This difference in phase-III behaviour can be explained by a consideration of the pheromone adjustment operations of each algorithm. For ACO, convergence cannot only be defined in the context of the distribution of solutions throughout the solution space (*i.e.* the point at which $d_\Sigma(t) = 0$), but also in a pheromone value context. That is, an ACO algorithm has converged when the pheromone value on all paths, except for a single path $S \in \mathbb{S}$, is effectively zero (*i.e.* zero for all computational purposes). At such a point, ants will only select edges from path S. For both AS and AS_{elite}, as pheromone values of paths become more dominant, the natural positive feedback process of the colony's path selection will dictate that the pheromone value on all edges, other than that of the increasingly dominant path, will decay to zero. Thus, these algorithms will converge to the point where $d_\Sigma(t) = 0$. However, both AS_{rank} and MMAS contain mechanisms that moderate this positive feedback process. Firstly, in the update process for AS_{rank}, in addition to the elitist ants, there are σ-1 unique paths that receive a weighted pheromone addition within each iteration. What this means for AS_{rank} is that there are always multiple paths for which the pheromone value does not decay to zero. Within MMAS, the pheromone bounding ensures that the pheromone values on all paths do not go below $\tau_{min}(t)$.

The search-time statistics in Figure 2(c) (the triangle and dot lines plots superimposed over the bar charts) indicate the range of iteration numbers in which each algorithm found S_{gb}, the global best solution for the run. Interestingly, all four algorithms tended to find their global best solutions towards the end of phase-I and the beginning of phase-II, albeit MMAS at a later stage than the other three algorithms. AS_{elite} and AS had a greater variation in their search-times than AS_{rank}, with MMAS having the greatest variation in its search-times.

6.2 Case Study 2: New York Tunnels Problem

6.2.1 Preliminaries

The New York Tunnels Problem (NYTP) was first considered by Schaake and Lai (1969) while Dandy *et al.* (1996) was the first to apply an evolutionary algorithm to this problem. The network is a gravity system fed from a single reservoir, and consists of 20 nodes connected via 21 tunnels (Figure 3). There is a single demand case for the problem. Each tunnel has a null option, or the option to provide a duplicate tunnel with one of 15 different diameter sizes. The reader is referred to Dandy *et al.* (1996) for the case study details. This case study is the second smallest considered in this chapter, and has a search space of approximately 1.934×10^{25} possible combinations.

6.2.2 Results

Based on the heuristics given in Table 1 $\{\tau_0, m\} = \{140, 90\}$ and based on preliminary analyses $I_{max} = 500$ was found to be sufficient. A single run of the NYTP consisted of 45,000 function

evaluations. The range of parameters tested was: $\sigma \in [2, 80]$ for AS_{elite}; $\sigma \in [2, 80]$ for AS_{rank}; $\{P_{best}, \delta\} \in [1 \times 10^{-5}, 0.99] \times [0, 0.99]$ for MMAS. For $2 \leq \sigma \leq 20$ the performance of AS_{elite} varied less than 1%, but for $\sigma > 20$ the solution quality was increasingly worse. For AS_{rank}, the performance varied less than 1% for the entire parameter range, with the better values being $8 \leq \sigma \leq 12$. For MMAS, the performance varied less than 1% for $0.005 \leq P_{best} \leq 0.99$ and $\delta \leq 0.0005$, with the solution quality degrading for lower values of P_{best} and higher values of δ. The optimal parameter settings were as follows: $\sigma = 8$ for AS_{elite}; $\sigma = 8$ for AS_{rank}; $\{P_{best}, \delta\} = \{0.05, 5\text{x}10^{-5}\}$ for MMAS.

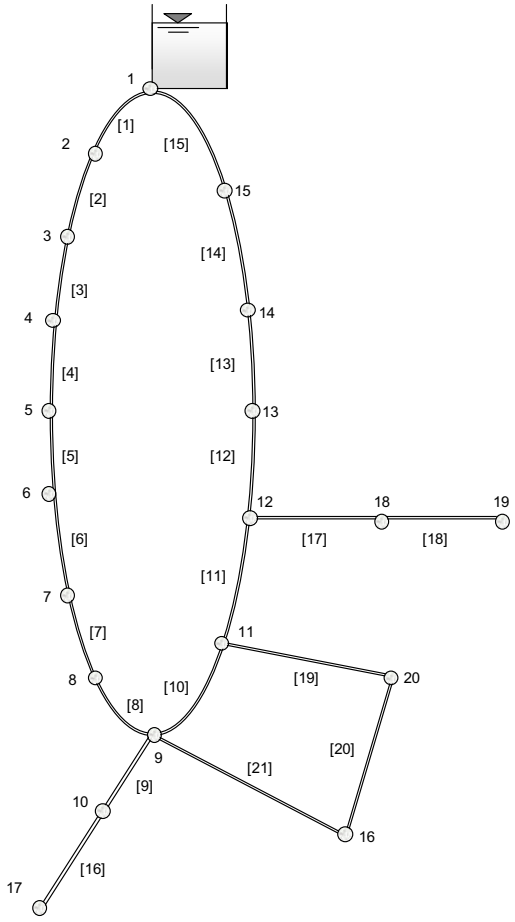

Figure 3. Network layout for New York Tunnels Problem

Table 3 gives a comparison of the performance of the ACO algorithms considered in this paper with that of the current best performing algorithms from the literature for the NYTP. A detailed statistical analysis of these algorithms was given in Zecchin *et al*. (2007), but all algorithms performed well, with AS_{elite}, AS_{rank} and MMAS, on average, finding solutions within a 1% cost of the known-optimum.

Algorithm	Best-cost ($M) *(% deviation from optimum)*						Mean search-time (evaluation no.)
	Minimum		Mean		Maximum		
AS	39.204	*(1.47)*	39.910	*(3.29)*	40.922	*(5.91)*	34,877
AS_{elite}	38.638	*(0.00)*	38.988	*(0.91)*	39.511	*(2.26)*	21,945
AS_{rank}	38.638	*(0.00)*	38.777	*(0.36)*	39.221	*(1.51)*	19,319
MMAS	38.638	*(0.00)*	38.836	*(0.51)*	39.415	*(2.01)*	30,711
PE[a]	41.800	*(8.18)*	-		-		NA
GA_{imp}[b]	38.796	*(0.41)*	NA		NA		96,750
GA[c]	37.13[i]	-	NA		NA		~1,000,000
$ACOA_{i\text{-}best}$[d]	38.638	*(0.00)*	NA		NA		13,928
TS[e]	37.13[i]	-	NA		NA		~10,000
$AS_{i\text{-}best}$[f]	38.638	*(0.00)*	38.849	*(0.55)*	39.492	*(2.21)*	22,052
ACS[g]	38.638	*(0.00)*	39.629	*(2.57)*	41.992	*(8.68)*	23,972
GA_{adapt}[h]	38.638	*(0.00)*	38.770	*(0.34)*	39.07	*(1.12)*	15,680

[a] Partial enumeration (Gessler, 1982). [b] Improved GA that used a variable exponent in fitness scaling, an adjacency mutation operator, and Gray code representation (Dandy *et al.* 1996). [c] Genetic algorithm (Savic & Walters, 1997). [d] Iteration-best updating version of ACO (Maier, *et al.* 2002). [e] Tabu search (Cunha & Ribeiro, 2004). [f] An improved iteration-best version of AS (Zecchin *et al.* 2005). [g] Ant colony system (Zecchin *et al.* 2007). [h] Parameter free, self-adapting, boundary searching genetic algorithm (Afshar & Marino, 2007). [i] Not assessed as feasible by EPANET2 (Maier *et al.*, 2002).

Table 3. Comparison of performance of AS, ACS, AS_{elite}, AS_{rank}, MMAS, and other algorithms from the literature applied to the New York Tunnels Problem. Results for AS, AS_{elite}, AS_{rank}, and MMAS are based on 20 runs. NA means that the information was not available

Plots of the iteration best-costs $f_{min}(t)$, the mean-colony-distance $d_{\Sigma}(t)$, and the searching phases and search-time statistics for the algorithms applied to the NYTP are given in Figure 4(a)-(c). Again, the three distinct searching phases observed for the TRP are observed in the behaviour of f_{min} and d_{Σ}. The relative behaviours of the algorithms applied to the NYTP are similar to that for the TRP, except for the faster convergence of AS in phase-I than that of both AS_{elite} and AS_{rank}. The effectiveness of the additional pheromone adjustment mechanisms in AS_{elite}, AS_{rank} and MMAS is made clear in Figure 4(a). This is seen by the fact that, for the majority of the phase-III searching, these algorithms have confined the search to