Ant Colony Optimisation Applied to Water Distribution System Design: A Comparative Study of Five Algorithms

Aaron C. Zecchin[1], Holger R. Maier[2], Angus R. Simpson[3], Michael Leonard[4], and John B. Nixon[5].

ABSTRACT

Water distribution systems (WDSs) are costly infrastructure, and much attention has been given to the application of optimisation methods to minimise design costs. In previous studies, Ant Colony Optimisation (ACO) has been found to perform extremely competitively for WDS optimisation. In this paper, five ACO algorithms are tested: one basic algorithm (Ant System) and four more advanced algorithms (Ant Colony System, Elitist Ant System, Elitist-Rank Ant System (AS_{rank}), and Max-Min Ant System (MMAS)). Experiments are carried out to determine their performance on four WDS case studies, three of which have been considered widely in the literature. The findings of the study show that some ACO algorithms are very successful for WDS design, as two of the ACO algorithms (MMAS and AS_{rank}) outperform all other algorithms applied to these case studies in the literature.

Keywords: *Metaheuristics; Ant Colony Optimisation; Water Distribution Systems; Optimisation*

[1] Postgraduate student, Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Adelaide, South Australia, Australia, 5005.

[2] Associate Professor, Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Adelaide, South Australia, Australia, 5005

[3] Associate Professor, Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Adelaide, South Australia, Australia, 5005

[4] Postgraduate student, Centre for Applied Modelling in Water Engineering, School of Civil and Environmental Engineering, The University of Adelaide, Adelaide, South Australia, Australia, 5005.

[5] Senior Research Scientist, Research and Development, United Water International Pty Ltd, Parkside, South Australia, Australia, 5063.

1 INTRODUCTION

Much research over the last 25 years has been dedicated to the development of techniques to minimise (optimise) the capital costs associated with water distribution system (WDS) infrastructure. Within the last decade, many researchers have shifted the focus of WDS optimisation from "traditional" techniques, based on linear and non-linear programming, to the implementation of "heuristics derived from nature" (HDNs) (Colorni *et al.* 1996), namely: genetic algorithms (GAs) (Simpson *et al.* 1994), simulated annealing (Cunha & Sousa 1999), and ant colony optimisation (ACO) (Maier *et al.* 2003). ACO is a HDN based on the foraging behaviour of ants (Dorigo *et al.* 1996). It has seen wide and successful application to many different optimisation problems (Dorigo & Gambardella 1999) and has recently been found to perform very competitively for WDS optimisation (Zecchin *et al.* 2005). Changes have been made to the initial and most simple formulation of ACO, Ant System (Dorigo *et al.* 1996), to improve the operation of the decision policy. The resulting algorithms provide different techniques for managing the trade-off between the two conflicting search attributes of exploration (the ability of the algorithm to search large areas of the solution-space) and exploitation (the ability of the algorithm to search more thoroughly near areas where good solutions have been found previously). These algorithms include: Ant Colony System (ACS) (Dorigo & Gambardella 1997); Elitist Ant System (AS_{elite}) (Dorigo *et al.* 1996); Elitist-Rank Ant System (AS_{rank}) (Bullenheimer *et al.* 1999); and Max-Min Ant System (MMAS) (Stützle & Hoos 2000). In this paper, these five algorithms (AS, ACS, AS_{elite}, AS_{rank}, and MMAS) are applied to four WDS problems. The objective is to assess the performance of these algorithms and determine which are best suited for WDS optimisation.

2 ANT COLONY OPTIMISATION

Over a period of time an ant colony is able to determine the shortest path from its nest to a food source. This perceived 'swarm intelligence' is achieved via an indirect form of communication between the colony members that involves them depositing and following a decaying trail of chemical substance, called pheromone, on the paths they travel. Over time, shorter paths are reinforced with greater amounts of pheromone, as they require less time to be traversed, thus becoming the dominant paths for the colony to follow. As a combinatorial optimisation algorithm, ACO is based on this analogy of the incremental learning of a colony by an iterative trial and error process. In the ACO algorithm, artificial ants construct solutions to the underlying combinatorial problem by probabilistically selecting options at each decision point. The probabilistic decision policy is governed by two weighting factors: one is the pheromone intensity (symbolised by τ), which is representative of the learned information; and the other is desirability (symbolised by η), which acts as a bias against higher cost options. At each iteration (i.e. generation of a new set of solutions by the colony), information from the previous iteration is used to alter the pheromone values and hopefully increase the probability of the optimum solution being found. In order to effectively use more recent information, the pheromone values are decayed with time (mimicking the evaporation of their real life counter part), thus placing more of an emphasis on recent information. For a more comprehensive formulation of ACO as a generalised metaheuristic, the reader is referred to Dorigo and Gambardella (1999).

2.1 THE ACO ALGORITHMS

Ant System (AS) is the original and simplest ACO algorithm (Dorigo *et al.* 1996). The decision policy used within AS is a probability function based on the relative weighting of pheromone intensity and desirability of each option at a decision point. It is parameterised by two parameters, α and β, which indicate the relative importance of pheromone intensity and desirability, respectively, in the decision process. At the end of each iteration, each of the m ants adds pheromone to their path (set of selected options). The amount of pheromone added is inversely proportional to the objective function value of the path (i.e. for minimisation problems, lower cost solutions are better, hence they receive more pheromone). The pheromone updating process is parameterised by three parameters: τ_0, the initial pheromone value; Q, a scaling factor for the pheromone additions; and ρ, the pheromone persistence factor, which is indicative of the relative importance of previous information ($0 < \rho < 1$).

In an attempt to regulate the trade-off between exploitation of the current best solution and further exploration of the solution space, Dorigo and Gambardella (1997) presented ACS. ACS includes additional rules that probabilistically determine whether an ant is to act in an exploitative or explorative manner at each decision point (determined by a parameter $0 \leq q_0 \leq 1$). Another mechanism used within ACS is the "local" updating of the pheromone of an ant's selected option immediately after it has generated its solution. This degradation discourages the re-selection of edges within an iteration and works to balance the exploitative decision policy by further encouraging exploration of alternate edges (governed by the local pheromone persistence ρ_l).

To exploit information about the current global-best solution, Dorigo *et al.* (1996) proposed the use of an algorithm known as AS_{elite}. This algorithm uses "elitist ants" (parameterised by σ, the number of elitist ants), which only reinforce the path of the current global-best solution after every iteration (analogous to elitism strategies used in GAs). The decision rule for AS_{elite} is the same as that for AS.

Proposed by Bullnheimer *et al.* (1999), AS_{rank} further develops the idea of elitism used in AS_{elite} to involve a rank-based updating scheme. At the end of an iteration, 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. This formulation effectively uses only the best information in a weighted manner as greater importance is given to the higher-ranking ants' solutions.

To overcome the problem of premature convergence whilst still allowing for exploitation, Stützle and Hoos (2000) developed 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 and a theoretically asymptotic upper limit. As a result of the lower bound 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. As the bounds serve to encourage exploration, provision for exploitation is made in MMAS by the addition of pheromone only to the iteration-best ant's path at the end of an iteration. To further exploit good information, the global-best solution is updated every T_{gb} iterations (T_{gb} = 10 is used here). 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 pheromone bounds and PTS are governed by parameters P_{best} and δ, respectively (Stützle and Hoos 2000).

2.2 APPLICATION OF ACO TO WATER DISTRIBUTION SYSTEM OPTIMISATION

The optimisation of WDSs can be formulated as a constrained minimisation problem (Zecchin *et al.* 2005). Essentially, WDS optimisation involves the selection of the lowest cost set of diameters for pipes within a network such that the design pressure constraints are not violated. As ACO, like most HDNs, cannot account for constraints explicitly, the WDS problem was converted into an unconstrained problem by use of a penalty function (Zecchin *et al.* 2005). Additionally, as in Maier *et al.* (2003), the desirability for an option was defined as the inverse of the cost of its implementation. As in Zecchin *et al.* (2005), a 'virtual-zero-cost' was used for options that had zero cost. The ACO program developed for this study, primarily coded in FORTRAN 90, used EPANET2 as the hydraulic solver.

3 CASE STUDIES AND RESULTS

The four case studies used in this research are: the Two Reservoir Problem (TRP); the New York Tunnels Problem (NYTP); the Hanoi Problem (HP); and the Doubled New York Tunnels Problem (2-NYTP). Selected details are given in Table 1, and the reader is referred to relevant references for further information (e.g. Simpson *et al.* (1994) for the TRP and Zecchin *et al.* (2005) for the others). The parameter settings used for the fundamental parameters α, β, ρ, τ_0, and m were based on the guidelines developed in Zecchin *et al.* (2005) with the following changes: for ACS, preliminary analysis showed that, due to a more complicated decaying behaviour arising from the dual effect of ρ and ρ_l, the Zecchin *et al.* (2005) heuristic for ρ was not applicable and hence ρ was calibrated; for AS_{elite} and AS_{rank}, the Zecchin *et al.* (2004b) heuristic for τ_0 was scaled up by σ. The remaining parameters for each algorithm were calibrated for each case study, based on extensive preliminary analyses. The resulting parameters are summarised in Table 1.

Table 2 contains a comparison of the results obtained using the five ACO algorithms considered with those obtained from a selection of other best performing algorithms for the four case studies. It is important to note that other authors have proposed lower cost solutions to the NYTP and HP (Savic & Walters 1997; Lippai *et al.* 1999; Cunha & Sousa 1999; Wu *et al.* 2001), however, these solutions were assessed as being infeasible by EPANET2, which was the benchmark hydraulic analysis tool used for this work. The performance of the algorithms is based on solution quality (i.e. *best-cost* meaning the minimum cost found in a run) and search efficiency (i.e. *search-time* meaning the number of function evaluations required to find the best-cost for each run). As the algorithms represent stochastic processes, statistics for the ACO algorithms are based on 20 runs (each with different random number generator seeds). Statistical *t*-tests are performed to determine the significance of the deviation of the algorithms' mean best-cost from the optimum or the known best-cost (depending on whether the optimum is known or not).

For the TRP, all considered ACO algorithms found the optimum solution for all runs, except ACS, whose mean performance was statistically significantly different from the optimum. AS_{elite} and AS_{rank} were more efficient than all of the other algorithms, including $ACOA_{i\text{-}best}$ (Maier *et al.* 2003), which was the current best algorithm for solving the TRP. The influence of the exploration encouraging mechanisms of MMAS is reflected by its relatively long search-times. AS, AS_{elite}, AS_{rank}, and MMAS performed better than GA_{prop} and ACOA, based on all measures of solution quality and efficiency. The four stated ACO algorithms yielded a similar quality performance to GA_{tour} and $ACOA_{i\text{-}best}$, but more efficiently. Based on these results, AS_{rank}, AS_{elite}, AS, and MMAS yield the current best and most efficient performance (in the stated order) for the TRP in the literature. As this case study represents a relatively small problem, it is seen that the increased exploitative mechanisms of AS_{elite} and AS_{rank} result in the increased efficiency of these algorithms without a reduction in solution quality.

As can be seen from Table 2, AS was the only tested ACO algorithm unable to find the known-optimum for the NYTP. ACS, AS_{elite}, AS_{rank}, and MMAS all found the known-optimum solution but only MMAS's mean best-cost was not significantly different to that of AS_{rank}, the best performing algorithm for this case study. As MMAS performed better than AS_{elite}, it can be deduced that, as opposed to the smaller TRP, the shift in emphasis away from exploitation yields an improved performance for MMAS for this case study. The greater emphasis on exploration in the pheromone update scheme of AS_{rank} (i.e. the inclusion of the top $\sigma - 1$ ranking ants, as opposed to only the top "elitist" ant) can be seen as a better-suited compromise between exploration and exploitation. Again, the relationship between exploitation and search-time is seen, in that the exploitive algorithms (AS_{elite} and AS_{rank}) have relatively shorter search-times. In comparison with the other algorithms from the literature, four of the tested ACO algorithms perform better than GA_{imp} in terms of both solution quality and efficiency. Only the minimum cost found by $ACOA_{i\text{-}best}$ is given, but it is known that $ACOA_{i\text{-}best}$ did not find the optimum in all runs. AS_{rank} and MMAS were able to produce a better average performance than $AS_{i\text{-}best}$, the current best performing algorithm for the NYTP. Based on its greater robustness and efficiency, AS_{rank} produces the current best performance for the NYTP.

For the HP, it can be seen in Table 2 that AS was unable to find a single feasible solution in any of the 20 runs. The difficulty of finding feasible solutions for this problem has been recognised by other authors (Eusuff & Lansey 2003), and is mainly attributed to the relatively small feasible region of the search space (Zecchin *et al.* 2005). ACS was able to find feasible, albeit poor, solutions. AS_{elite} was able to find better solutions than ACS, however, they were still relatively poor. AS_{rank} was able to find relatively good solutions. MMAS was found to be the best performing algorithm for this case study, as it was able to find a new lowest cost feasible solution, 0.78% less than the previous lowest cost solution, found by fmGA1 (Wu *et al.* 2001) (this result is in accordance with the findings of Zecchin *et al.* (2006): the reader is referred to that paper for the solution details). MMAS achieved the lowest mean best-cost from the known optimum (the *t*-tests showed that the performance of MMAS was significantly different to that of all the other tested ACO algorithms), but also had the longest mean search-time (but still shorter than that of the GAs). In general, the performances of ACS, AS_{elite}, AS_{rank}, and MMAS were much more sensitive to their respective parameter settings for this case study, such that only moderate variations from the selected parameters resulted in the inability to find feasible solutions for some runs. The best parameter settings for this case study vary greatly from those of all the other case studies. A common thread is that the optimal parameter settings for this case study increased each of the algorithm's emphasis on exploitation. For example: for ACS, $q_0 = 0.6$, which means that 60% of decisions made were exploitative as opposed to 0% for the other case studies ($q_0 = 0$); for AS_{elite}, the number of elitist ants for this case study was far greater than for the other case studies; and for MMAS, the fact that P_{best} had a greater value indicates looser pheromone bounding and consequently reduced exploration potential (similarly, δ being set to a low value also indicates a reduction in MMAS's exploration potential). Despite this notable sensitivity, the parameter heuristics proposed by Zecchin *et al.* (2005) resulted in extremely good performance for MMAS and, to a lesser extent, AS_{rank}. The observed importance of exploitation, for searching in the infeasible region, can be attributed to the necessity of the algorithm to effectively use information provided by the best, albeit infeasible, solutions to lead the search to the feasible region. This fact also highlights the importance of the penalty function to give an accurate indication of the distance of a solution from the feasible region.

For the NYTP-2, only ACS, AS_{rank}, and MMAS were able to find the optimum. AS_{elite} was unable to perform as well as AS_{rank} (and MMAS), despite its longer average search-time. The increased exploration capability of MMAS yielded an improved performance for this larger case study. This emphasis on exploration is reflected in the greater search time of MMAS (over 2.5 times greater than that of AS_{elite} and over 3 times greater than that of AS_{rank}). In comparison with $AS_{i\text{-}best}$, only MMAS was able to achieve a lower mean best-cost, but at a far longer mean search-time. As MMAS achieved a better mean than $AS_{i\text{-}best}$, it provides the current best performance for the 2-NYTP within the literature. The *t*-tests showed that MMAS performed significantly better than all of the other tested ACO algorithms, except AS_{rank}.

A comparison of the results of the experiments for the case studies shows that a greater emphasis on exploitation is important for the smaller case studies and a greater emphasis on exploration for the larger case studies. The implications of these findings are that an algorithm that encourages exploration will perform better for larger case studies, as a greater spread of candidate solutions will be achieved across the search space. However, this behaviour can have adverse effects, in terms of computational efficiency, for smaller case studies, where a more focused search process, utilizing exploitation, performs more efficiently.

Table 3 provides a ranking of each algorithm's performance for all four case studies. AS (ranked fourth) performed extremely well for the smallest case study, but was the worst performing algorithm for the larger case studies. ACS (ranked last) performed significantly worse than all other algorithms for the smallest case study, and performed relatively moderately for the other case studies (with a consistently high variability in its solution quality). AS_{elite} (ranked third) performed consistently well for all case studies and achieved the second highest efficiency ranking. AS_{rank} (ranked second) was the most efficient algorithm and performed best (or statistically equivalent to the best) for all case studies except for the HP. AS_{rank}'s performance for the TRP and NYTP are the best found in the literature, subject to hydraulic feasibility determined by EPANET2. MMAS (ranked first), despite being ranked second to last for all case studies in terms of efficiency, provided the best (or statistically equivalent to the best) performance for all case studies and was the only algorithm to find the best known optimum for the HP. MMAS's performances for the HP and 2-NYTP are the best found in the literature, subject to hydraulic feasibility determined by EPANET2.

4 SUMMARY AND CONCLUSIONS

Five Ant Colony Optimisation (ACO) algorithms have been applied to four water distribution system (WDS) optimisation problems: the Two-Reservoir Problem (TRP); the New York Tunnels Problem (NYTP); the Hanoi Problem (HP); and the Doubled New York Tunnels Problem (2-NYTP). Four of the algorithms—Ant Colony System (ACS), Elitist Ant System (AS_{elite}), Elitist-Rank Ant System (AS_{rank}), and Max-Min Ant System (MMAS)—are current state-of-the-art ACO algorithms that have been applied successfully to a variety of combinatorial optimisation problems. The other algorithm—Ant System (AS)—is the most basic and original form of ACO. The parameter guidelines given in Zecchin *et al.* (2005) were used effectively, where appropriate, for all algorithms tested.

The results obtained indicate that, AS_{rank} and MMAS stand out from the other ACO algorithms in terms of their consistently good performances. Compared with MMAS, AS_{rank} was more efficient but it is important to note that AS_{rank} did not perform as well as MMAS for the larger, and hence more difficult, case studies. The better performance of MMAS with these case studies could be largely attributed to its greater ability to explore (resulting, however, in longer search-times). The results, based on only four case studies from the literature, are extremely promising, but a wider experimentation of ACO algorithms applied to more case studies is required to determine their utility for real world WDS design problems.

5 ACKNOWLEDGMENT

The authors thank Dr. Stephen Carr, Mr. Andrew J. Roberts, and Mr. Matthew J. Berrisford for their work in the development and simulation phase, and The University of Adelaide and United Water International Pty. Ltd. for their financial assistance.

6 REFERENCES

Bullnheimer, B., Hartl, R.F. & Strauss, C. (1999). “A new rank based version of the Ant System: A computational study.” *Central European Journal for Operations Research and Economics*, 7(1), 25–38.

Colorni, A., Dorigo, M., Maffoli, F., Maniezzo, V., Righini, G. & Trubian, M. (1996). “Heuristics from nature for hard combinatorial optimisation problems.” *International Transactions in Operational Research*, 3(1), 1–21.

Cuhna, M.C. & Sousa, J. (1999). “Water distribution network design optimization: Simulated annealing approach.” *Journal of Water Resources Planning and Management*, ASCE, 125(4), 215–221.

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

Dorigo, M., Di Caro, G. & Gambardella, L.M. (1999). “Ant algorithms for discrete optimisation.” *Artificial Life*, 5(2), 137–

172.

Dorigo, M. & Gambardella, L.M. (1997). “Ant colony system: A cooperative learning approach to TSP.” *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66.

Dorigo, M., Maniezzo, V. & Colorni, A. (1996). “The ant system: Optimisation by a colony of cooperating agents.” *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, 26(1), 29–41.

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

Lippai, I., Heany, J.P. & Laguna, L. (1999). “Robust water system design with commercial intelligent search optimizers.” *Journal of Computing in Civil Engineering*, ASCE, 13(3), 135–142.

Maier, H.R., Simpson, A.R., Zecchin, A.C., Foong, W.K., Phang, K.Y., Seah, H.Y. & Tan, C.L. (2003). “Ant colony optimisation for the design of water distribution systems.” *Journal of Water Resources Planning and Management*, ASCE, 129(3), 200–209.

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

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

Simpson, A.R. & Goldberg, D.E. (1994). “Pipeline optimization via genetic algorithms: From theory to practice.” *Proc. 2nd Int. Conf. On Pipeline Systems*, Edinburgh, Scotland, May, 309–320.

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

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

Zecchin, A.C., Maier, H.R., Simpson, A.R., Leonard, M., Roberts, A.J. & Berrisford, M.J. (2006) “Application of two ant colony optimisation algorithms to water distribution system optimisation.” *Mathematical and Computer Modelling*, In press.

Zecchin, A.C., Simpson, A.R., Maier H.R. & Nixon, J.B. (2005). “Parametric study for an ant algorithm applied to water distribution system optimization.” *IEEE Transactions on Evolutionary Computation*, 9(2), 175-191.

7 TABLE CAPTIONS AND TABLES

Table 1 Case study details and calibrated parameter values for the ACO algorithms.

Table 2 Comparison of performance of AS, ACS, AS_{elite}, AS_{rank}, MMAS, and other algorithms from the literature applied to the four case studies. Results for AS, ACS, AS_{elite}, AS_{rank}, and MMAS are based on 20 runs. Note: NA means that the information was not available, N/A means not applicable, NFS means no feasible solution was found, and STD means standard deviation.

Table 3 Relative ranking and performance summary in terms of solution quality (i.e. mean best-cost) and efficiency (i.e. mean search-time) of the five ACO algorithms AS, ACS, AS_{elite}, AS_{rank}, and MMAS applied to the four WDS case studies TRP, NYTP, HP, and 2-NYTP. The total score is the sum of the ranks for each case study and the overall rank for each algorithm is ordered in ascending order of the total score.

Table 1 Case study details and calibrated parameter values for the ACO algorithms.

Case study	No. of pipes	No. of nodes	Search space size	I_{max}	ACS $\{\rho, \rho_l, q_0\}$	AS_{elite} σ	AS_{rank} σ	MMAS $\{P_{best}, \delta\}$
TRP	14	32	3.78×10^{7}	400	{0.9, 0.98, 0}	4	10	$\{0.5, 10^{-6}\}$
NYTP	20	21	1.93×10^{25}	500	{0.98, 1, 0}	8	8	$\{0.05, 5 \times 10^{-5}\}$
HP	34	32	$\sim 2.87 \times 10^{26}$	1,500	{0.8, 0.7, 0.6}	40	20	{0.5, 0}
2-NYTP	40	41	3.741×10^{50}	3,000	{0.999, 1, 0}	3	8	{0.001, 0}

Table 2 Comparison of performance of AS, ACS, AS_{elite}, AS_{rank}, MMAS, and other algorithms from the literature applied to the four case studies. Results for AS, ACS, AS_{elite}, AS_{rank}, and MMAS are based on 20 runs. Note: NA means that the information was not available, N/A means not applicable, NFS means no feasible solution was found, and STD means standard deviation.

Case study	Algorithm	Best-cost ($M) *(% deviation from optimum)*				Comparison of mean best-cost with optimum or best algorithm[a]		Search-time (evaluation number $\times 10^3$)	
		Minimum	Mean	Maximum	STD	*t* statistic[b]	*(P-value)*	Mean	STD
TRP	AS	1.750 *(0.00)*	1.750 *(0.00)*	1.750 *(0.00)*	0.000	N/A		2.1	0.6
	ACS	1.750 *(0.00)*	1.770 *(1.13)*	1.904 *(8.81)*	0.050	1.774[c]	*(0.046)*	5.0	2.3
	AS_{elite}	1.750 *(0.00)*	1.750 *(0.00)*	1.750 *(0.00)*	0.000	N/A		1.8	0.8
	AS_{rank}	1.750 *(0.00)*	1.750 *(0.00)*	1.750 *(0.00)*	0.000	N/A		1.5	0.5
	MMAS	1.750 *(0.00)*	1.750 *(0.00)*	1.750 *(0.00)*	0.000	N/A		3.0	0.9
	GA_{prop}[d]	1.750 *(0.00)*	1.759 *(0.51)*	1.812 *(3.54)*	0.020	1.335	*(0.106)*	23.6	13.9
	GA_{tour}[e]	1.750 *(0.00)*	1.750 *(0.00)*	1.750 *(0.00)*	0.000	N/A		8.7	NA
	ACOA[f]	1.750 *(0.00)*	1.769 *(1.09)*	1.813 *(3.60)*	0.030	1.964[c]	*(0.039)*	12.5	NA
	$ACOA_{i\text{-}best}$[f]	1.750 *(0.00)*	1.750 *(0.00)*	1.750 *(0.00)*	0.000	N/A		8.5	NA
NYTP	AS[g]	39.204 *(1.47)*	39.910 *(3.29)*	40.922 *(5.91)*	0.552	8.413[c]	*$(O\{10^{-10}\})$*	35.9	6.7
	ACS	38.638 *(0.00)*	39.629 *(2.57)*	41.992 *(8.68)*	0.803	4.488[c]	*$(O\{10^{-10}\})$*	24.0	5.6
	AS_{elite}	38.638 *(0.00)*	38.988 *(0.91)*	39.511 *(2.26)*	0.323	2.415[c]	*(0.021)*	21.9	3.2
	AS_{rank}	38.638 *(0.00)*	38.777 *(0.36)*	39.221 *(1.51)*	0.200	N/A		19.3	2.4
	MMAS[g]	38.638 *(0.00)*	38.836 *(0.51)*	39.415 *(2.01)*	0.305	0.694	*(0.492)*	30.7	6.0
	GA_{imp}[h]	38.796 *(0.41)*	NA	NA	NA	NA		96.8[g]	NA
	$ACOA_{i\text{-}best}$[f]	38.638 *(0.00)*	NA	NA	NA	NA		13.9	8.2
	$AS_{i\text{-}best}$[i]	38.638 *(0.00)*	38.849 *(0.55)*	39.492 *(2.21)*	NA	NA		22.0	NA
HP	AS[g]	NFS	NFS	NFS	-	-	-	-	-
	ACS	7.754 *(26.41)*	8.109 *(32.20)*	8.462 *(37.96)*	0.198	32.191[c]	*$(O\{10^{-29}\})$*	61.3	34.3
	AS_{elite}	6.827 *(11.30)*	7.295 *(18.93)*	8.187 *(33.48)*	0.346	10.709[c]	*$(O\{10^{-13}\})$*	59.9	9.6
	AS_{rank}	6.206 *(1.17)*	6.506 *(6.07)*	6.788 *(10.66)*	0.150	2.527[c]	*(0.016)*	75.3	12.1
	MMAS[g]	6.134 *(0.00)*	6.394 *(4.24)*	6.635 *(8.17)*	0.122	N/A		85.6	22.3
	GA No. 2[j]	6.195 *(1.00)*	NA	NA	NA	NA		10^3[k]	NA
	fmGA1[l]	6.182 *(0.78)*	NA	NA	NA	NA		113.6[k]	NA
	$AS_{i\text{-}best}$[i]	6.367 *(3.80)*	6.842 *(11.54)*	7.474 *(21.95)*	NA	NA		67.1	NA
2-NYTP	AS	80.855 *(4.63)*	83.572 *(8.15)*	85.267 *(10.34)*	1.291	16.788[c]	*$(O\{10^{-19}\})$*	131.8	19.2
	ACS	77.275 *(0.00)*	80.586 *(4.28)*	86.682 *(12.17)*	2.656	3.822[c]	*$(O\{10^{-4}\})$*	472.0	30.2
	AS_{elite}	77.922 *(0.84)*	79.806 *(3.28)*	81.986 *(6.10)*	1.248	5.141[c]	*$(O\{10^{-6}\})$*	90.4	34.7
	AS_{rank}[m]	77.434 *(0.21)*	78.492 *(1.58)*	79.863 *(3.35)*	0.599	1.538	*(0.132)*	72.3	12.6
	MMAS	77.275 *(0.00)*	78.213 *(1.21)*	79.353 *(2.69)*	0.518	N/A		238.3	122.9
	$AS_{i\text{-}best}$[i]	77.275 *(0.00)*	78.302 *(1.33)*	79.922 *(3.43)*	NA	NA		75.8	NA

[a]Tests are made against optimum value for the TRP, AS_{rank} mean best-cost for NYTP, and MMAS mean best-cost for HP and 2-NYTP. [b]For the TRP, test is a one sided *t*-test for H_0: mean best-cost > optimum, test based on 19 degrees of freedom for AS, ACS, AS_{elite}, AS_{rank}, and MMAS and 9 for others, and for the NYTP, HP and 2-NYTP, test is an unpaired, two-sample *t* statistic with 38 degrees of freedom for H_0: the respective algorithm's mean best-cost is equivalent to that of AS_{rank} (for NYTP) and MMAS (others). [c]Significant at a 5% significance level for the relevant test. [d]Simpson *et al.* (1994). [e]Simpson & Goldberg (1994). [f]Maier *et al.* (2003). [g]The results for these algorithms are in agreement with Zecchin *et al.* (2004a), who conducted similar trials. [h]Dandy *et al.* (1996). [i]Zecchin *et al.* (2005). [j]Savic & Walters (1997). [k]Only value reported. [l]Wu *et al.* (2001). [m]AS_{rank} was able to find the known-optimum solution for other parameter settings but at a poorer mean best-cost.

Table 3 Relative ranking and performance summary in terms of solution quality (i.e. mean best-cost) and efficiency (i.e. mean search-time) of the five ACO algorithms AS, ACS, AS_{elite}, AS_{rank}, and MMAS applied to the four WDS case studies TRP, NYTP, HP, and 2-NYTP. The total score is the sum of the ranks for each case study and the overall rank for each algorithm is ordered in ascending order of the total score.

Case Study	ACO Algorithms relative rank in terms of Solution quality and (*efficiency)*				
	AS	ACS	AS_{elite}	AS_{rank}	MMAS
TRP	1[b, c, e] *(3)*	5[c] *(5)*	1[b, c, e] *(2)*	1[a, c, e] *(1)*	1[b, c, e] *(4)*
NYTP	5 *(5)*	4[c] *(3)*	3[c] *(2)*	1[a, c, e] *(1)*	1[b, c, e] *(4)*
HP	5[d] *(5)*	4 *(3)*	3 *(1)*	2 *(2)*	1[a, c] *(4)*
2-NYTP	5 *(3)*	4[c] *(5)*	3 *(2)*	1[b, c, e] *(1)*	1[a, c, e] *(4)*
Total	16 *(16)*	17 *(16)*	10 *(7)*	5 *(6)*	4 *(16)*
Overall rank	4 *(3)*[e]	5 *(3)*[e]	3 *(2)*	2 *(1)*	1 *(3)*[e]

[a]Algorithm yields best performance in literature for given case study subject to feasibility determined by EPANET2. [b]Algorithm yields statistically equivalent performance to best performing algorithm. [c]Optimum or known-optimum was found. [d]No feasible solutions were found. [e] Numerous algorithms "drew" for this rank.

INTENTIONALLY BLANK