Ant Colony Optimisation for Large-Scale Water Distribution Network Optimisation

Laura Baker[1]**, Ed Keedwell**[1]**, Mark Randall-Smith**[2]

Abstract. In this paper we show that ant colony optimisation (ACO) can be successfully applied to large-scale water distribution network optimisation problems. In addition, a new ACO algorithm ERMMAS is proposed and is tested on this problem. A water distribution network taken from industry is optimised by a number of ACO systems and a well-tuned GA. The results indicate that although there are not large-scale efficiency savings to be made, ACO is capable of finding results of equal or better optimality than a comparable GA.

1 INTRODUCTION

Water distribution networks (WDN) serve to transport clean water from treatment works to individual customers and usually represent a significant capital investment in the development of the urban environment. The problem of designing a WDN to optimally meet performance criteria, such as delivering sufficient water pressure for high rise buildings and fire fighting; whilst minimising cost criteria, such as the cost of material, excavation, frequency of maintenance is known to be NP hard. A large variety of computational algorithms have been devised for this task which include well known techniques in operational research such as linear, dynamic and integer programming. In recent years however, a variety of nature-inspired and meta-heuristic algorithms such as genetic algorithms, simulated annealing and tabu search have been widely investigated as useful research tools for WDN design. Amongst these meta-heuristic algorithms, genetic algorithms (GAs) has proved to be one of the most popular with the application of GAs to WDN optimisation tracing back to the mid-nineties (Dandy et al., 1996; Savic and Walters, 1997). Whilst GAs have provided good solutions to water distribution optimisation problems for some time, the steady increase in the complexity of the network information being kept by the water companies means that GAs are no longer always suitable. This is in part due to the long running times incurred by the algorithm due and in particular, the high number of objective function evaluations required by evolutionary techniques. An increasing number of elements in the network and more detailed 24-hour simulation studies has seen the complexity of a single network simulation increase massively. Therefore, researchers are constantly looking for techniques which might deliver GA-class results, but with fewer objective function calculations. In this paper we investigate the application of a swarm intelligent approach to the problem of water distribution network optimisation. We describe the application of three ant colony systems to a large-scale water distribution network taken from industry and compare it with a well-tuned GA, with mixed results. The remainder of this section discusses water distribution network optimisation and previous research into using ant colonies for this purpose.

[1] School of Engineering, Computing and Mathematics, University of Exeter, Harrison Building, North Park Road, Exeter, UK, EX4 4QF Email: {lb281, E.C.Keedwell}@ex.ac.uk
[2] Mouchel, Priory Court, Poulton, Cirencester, Glocs, UK GL7 5JB Email: mark.randall-smith@mouchel.com.

Water Distribution Networks

WDNs are part of the water supply system, comprising of number of interconnected elements such as pipes, nodes, pumps, valves, and reservoirs of varying shapes and sizes. The nodes represent combined points of water demand (e.g. housing or industrial estates) on the system. The purpose of the network is to deliver water to the demand nodes from the water treatment works, reservoir, or other source throughout the day and under varying demand conditions. The demands on a WDN fluctuate throughout the day. Peak demands occur when people prepare to leave for work at around 7am until 9am and when industrial organisations begin work for the day. It is important that demands on the network at peak times are satisfied. However, WDNs are costly to construct, maintain and operate hence the need for the optimal design of WDNs, where least cost can be balanced with required water pressure levels.

There are many options to be considered when optimising a WDN, but in most case, an existing network is already in place making it difficult to attempt major structural change in the existing design. Changing the position of the network elements is considered a major structural change and would be very costly and therefore many studies are restricted to the rehabilitation of components within the network. This is achieved by replacing existing components which no longer meet the demands placed on the system with more suitable infrastructure. However, as this is a large capital investment, the water companies inevitably want these modifications to last for long time periods, typically 50-100 years. Therefore rehabilitation studies investigate changes such as replacing pipes, pumps and tanks with different sizes/specification of the same element can have a large effect on the performance of the network.

Attempting to find the optimum design for a WDN by hand is an arduous task, but engineers have done this in the past. AI techniques (e.g. the GA) make developing a proposed solution much quicker and easier although it can be difficult to align with engineering expectations. To perform an exhaustive search is implausible even with a small network which is demonstrated by a simple example known as the New York Tunnels network (Schaake and Lai, 1969) shown in Figure 1.

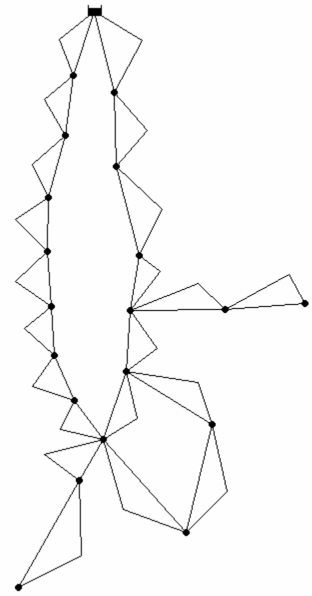

Figure 1 - Schematic of the New York Tunnels network. A single reservoir at the top feeds demand nodes located around New York via a set of duplicated pipes.

The optimisation algorithm must determine the pipe diameters of the 'new' pipes which represent expansions to the existing system. The algorithm has 16 possible commercial diameters to select from and there are 21 pipes to size, each with a potential effect on the other sizes in the network. Therefore, even in this small network there are 16^{21} or $1.93*10^{25}$ possible combinations of diameters to select from. This gives an indication of the problem complexity faced by the optimisation algorithm even for small problems.

The Example Network

The water distribution network used in this study is a real network taken from industry and represents the entire network for a large North American city. The attributes of the network taken from a hydraulic simulator are shown below:

Table 1 - Element counts of the example network

Element	**Number**
Number of Junctions	543
Number of Reservoirs	4
Number of Tanks	26
Number of Pipes	686
Number of Pumps	32
Number of Valves	39

Although not all of these elements will be subject to optimisation, the complexity involved with this network is clearly much greater than the simple network given above.

In this study, two different problems setups are considered, ranging from highly complex involving all elements of the network to smaller problems which only consider the pipe sizes required in the optimisation of the WDN.

The largest of the problems consists of some 161 decision variables which includes pipe sizes, pump and valve settings and a variety of tank expansion sizes. This variability in the number of decision variables makes the complexity a little more difficult to determine, but the number of options for the algorithms to consider is in the region of $6 *10^{82}$ for this largest problem.

Ant Colony Optimisation Approaches to WDN Optimisation

Ant colony optimisation (ACO) has been successfully applied to a wide range of optimisation problems (Dorigo and Di Caro) and has also been shown to perform very competitively for the optimisation of a WDN (Zecchin et al. 2003). Therefore, existing research shows that ACO and its variants could prove to be a suitable long term alternative to a GA.

According to Simpson et al (2003), ACO can be an appealing alternative to GA for the design of optimal WDNs. In this study ACO was applied to two benchmark WDN optimisation problems and in both scenarios ACO outperformed a GA in terms of computational efficiency and ability to find near global optimal solutions. Additionally Zecchin at al (2007) compared five variations of ACO for the purposes of water distribution network optimisation. The implementations in this study included one basic/standard implementation, an elitist ant system, an elitist-rank ant system (ERAS) and a max-min ant system (MMAS). The results showed that MMAS and ERAS outperformed all other algorithms that have been applied to the same four case studies. The four non standard implementations of ACO are 'current state-of-the-art ACO algorithms that have been applied successfully to variety of combinatorial optimisation problems' (Zecchin et al. 2007). They indicate that the consistently good performance of both ERAS and MMAS makes them stand out from the all other ACO algorithms. It is shown that ERAS is more efficient than MMAS for smaller case studies where as MMAS outperformed ERAS for larger case studies.

Therefore this paper investigates the application of ant colony optimisation to a real-world network taken from industry. The network is far larger than the New York Tunnels example and had previously been optimised using a GA which took several days to complete. The investigation will enable us to determine whether ant colony optimisation can be used to optimise the vastly increased search spaces associated with industrial water distribution networks.

2 METHOD

Infrastructure

The standard method for utilising GAs in WDN optimisation is to couple the GA with a hydraulic simulator such as the one used in this study, Epanet (Rossman, 1999). The GA generates a chromosome representing a candidate solution which is then evaluated in the normal way using the objective function. The objective function calls the DLL and simulates the solution in the Epanet and returns a number of computed results. The function then converts these results into fitness and penalty values and returns an overall fitness to the GA which can then proceed with algorithm.

In this study, we adopted a similar strategy whereby all of the calculations necessary for determining the objective function values were completed within a separate DLL, incorporating the hydraulic solver. The ant colony or any other optimisation algorithm simply passes the DLL the candidate solution and the DLL returns a single value indicating the fitness of that solution. By utilising this information-hiding method, the algorithm can be substituted quite easily with minimal recoding required.

Problem Setup

As described in the introduction, a number of different problems were considered in the study to determine the effect of changes in problem complexity on the algorithms involved. However, this only effects the decisions made by the algorithm. The objective function remained the same for all runs and was as follows:

Cost+G(PenaltyCost)

The G term is to balance the difference in magnitude between the two costs. The cost calculations are quite complex, but effectively each element that the algorithm selects to enhance the network has a known attached cost and these are simply summed together to give the total cost.

The penalty cost relates to the constraints placed on the network and therefore takes into account the following constraints:

- **Required Head Constraints:** All nodes in the network require a certain pressure 'head' to maintain service to customers. This constraint computes the difference between the actual pressure in the node and the required. If the pressure is too low, then the solution is penalised proportionately
- **Tank Level Constraints:** The solution is penalised if a tank drops below a pre-determined threshold or 'overspills'. Additionally, the tank must return to a certain percentage of its original level over 24 hours.
- **Velocity Constraints:** The solution is penalised if the velocity of the water passing through the pipes is too high (leading to an increased likelihood of leakage) or too low (leading to poor water quality).

These constraints are multiplied by constants to increase or decrease their importance to the algorithm and summed together to give the total penalty cost.

The setting of the constants for the objective functions is a non-trivial problem in itself, and the settings used in this study were the result of extensive experimentation with the GA. They were kept constant for both algorithms throughout this study.

Problem Setups

This study includes four problem setups they are as follows:

Setup 1 – considers 161 variables including pipes, pumps, tanks and valves. This is the largest problem and has around 10^{82} possible combinations

Setup 2 – similar to setup 1, but considers only pipe sizing and therefore 121 variables and a complexity in the region of 10^{71}.

ACO Methods

Basic Ant System

BAS calculates the probability of selecting a particular path at any given decision point according to the level of pheromone on that path and (optionally) some local heuristic values.

The calculated probability is then used in a probability proportionate roulette wheel which selects a path. The roulette wheel reinforces good solutions as the option with the highest probability of being selected has proportionately more chance of being selected on the wheel. The element of randomness involved in a roulette wheel encourages exploration and can help avoid stagnation. For more detail information on BAS readers are directed towards Dorigo (1996).

Max-Min Ant System

The MMAS was first proposed by Stützle and Hoos as a variant of the basic ant system. It has also been shown to be an attractive alternative to the GA (Bullenheimer et al, 1997). MMAS follows the same procedure for selecting a path as is described for BAS. MMAS differs to the basic ant system in the way in which the pheromone trails are updated. In MMAS pheromone is only added to one solution per iteration, where a combination of updating the iteration best solution (Sib) and the global best solution (Sgb) is utilised. A slight variation of the MMAS proposed in Stützle and Hoos (2000) is implemented. The difference occurs in the combination of updating the iterations *Sib* and *Sgb*. Similar ratios to Stützle and Hoos are used in this implementation but the change from using *Sib* to *Sgb* is marked by a percentage of the number of iterations opposed to actual values. In this implementation for the first 10% of iterations only the *Sib* is updated, from 10% to 30% update *Sgb* once per 6 iterations, from 30% to 70% update Sgb once per 4 iterations, from 70% update *Sgb* once per 3 iterations. Using the percentage of iterations as a ratio marker opposed to actual values allows the number of iterations to vary.

Elitist-Rank Max-min Ant System

ERMMAS is a new technique that incorporates elements from MMAS and elements from the elitist rank ant system (ERAS) (Bullenheimer et al, 1997). ERMMAS uses dynamic pheromone limits as described for MMAS and an elitist rank update scheme as used in ERAS. Once per iteration ERMMAS awards rank proportionate pheromone to the elite ant's solutions. The focus in ERMMAS shifts from iteration best update to global best update in the same way as MMAS. ERMMAS updates e iteration-elite solutions (Sie) and e global-elite solutions (Sge). ERMMAS offers more efficiency in the use of fitness evaluations than MMAS by updating e trails opposed to 1 trail per iteration. Two different solutions can be virtually identical in cost and penalty cost but comprise very different components. In this scenario MMAS will only reward the solution with the lowest cost ignoring the solution that is virtually as fit. The solution that is not rewarded may however lead to better quality solutions than the solution with the higher fitness. ERMMAS would reward both solutions and thus has a higher probability than MMAS of finding a solution with lower cost and penalty cost.

The pheromone addition equation for ERMMAS is:

$$\Delta\tau_{i(j)}^{(t)} = \begin{cases} \left(\frac{Q}{f(S^{ie}(t))} \right) / \; R & \text{if } i_{(j)} \in S^{ie}(t) \\ 0 & \text{Otherwise} \end{cases}$$

Where R = the rank of the solution; dividing the pheromone addition by the rank ensures that only the solution that is ranked as number 1 will receive the full amount of pheromone. The number of elitist ants e is calculated as a 10th of the population size.

3 RESULTS

Determining Parameter Settings

As with many algorithms, ACO relies on a number of parameter settings being correctly set for the algorithm to function well. In this study, the pheromone decay and population size parameters were experimented with to determine reasonable settings for these. However, the hydraulic solver requires some 0.6 seconds to complete a network evaluation, so these experiments were conducted with only 250 iterations of the ACO algorithms.

Pheremone Decay

The optimum value for the pheromone decay parameter ρ for the standard ant system is previously been shown to be 0.8 (Simpson et al. in 2003) over several case studies. Therefore, this is used as a starting point for experiments performed on BAS. Six experiments are performed with the ρ from 0.7 through to 0.95. These experiments showed that for this case study BAS performs best with ρ = 0.85.

Stützle and Hoos (200) showed that with MMAS the lower the ρ the quicker convergence occurs but with a higher likelihood of being in a local maximum and found ρ = 0.98 to be optimum for the traveling salesman problem. The example WDN problem is significantly larger than the travelling sales man problem used in that study and therefore due to the size of the problems at hand it was decided to set ρ = 0.99 to ensure the slowest convergence possible by applying minimum decay at each iteration. Later experiments were conducted to analyse the effect on the quality of solutions provided when varying ρ.

Population Size

A large population size is not very efficient with respect to the number fitness evaluations as a lot of fitness evaluations are conducted before the trails are updated and used. Several experiments were performed with BAS to test the effect of altering the population size. Four experiments were performed with the population sizes 10, 50, 100 and 1000. It was observed from these experiments that over a given number of fitness evaluations the difference caused by using different population sizes narrows to virtually nothing. A population size of 100 provided solutions with slightly lower cost and penalty cost than all other populations sizes tested however the difference is marginal. For all further experiments with BAS a population size of 100 is used.

A population size of 100 is used in experiments with MMAS however given that MMAS only adds pheromone to one solution per iteration this means that 99 fitness evaluations are not utilised. Experiments have been performed to test the efficiency of using a population size of 60 compared to 100. The results showed that on every occasion a population size of 100 found a final solution with lower cost and penalty cost than a population size of 60 found.

Optimisation Results

Setup 1

Figure 3 plots the total cost of the global best solution f(Sgb) for BAS, MMAS and the GA each time a new global best solution Sgb is found. MMAS initially requires more fitness evaluations than BAS to reach a certain level of fitness. The evolution in BAS tails off at around 80 000 fitness evaluations where a new global best solution is not found from this point forward.

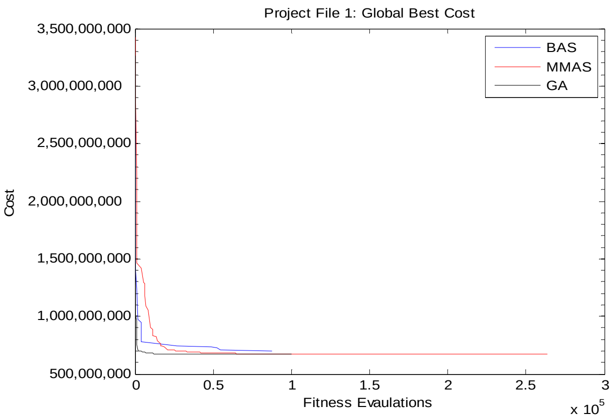

Figure 2 Fitness of the Global Best Solution for BAS, MMAS and the GA on Setup 1

In Figure 2 MMAS and BAS are both run for 300 000 fitness evaluations. MMAS continues to evolve the quality of solutions until over 200 000 fitness evaluations and shows evidence of further improvement. On the final iteration performed by MMAS the difference between the fitness of the iterations best solution f(Sib) and the mean average fitness of solutions f(Φ) was 921,083,790. The difference between the f(Φ) and f(Sib) indicates that further improvement on the quality of solutions could be achieved given more fitness evaluations as close f(Φ) and f(Sib) values indicate that stagnation is occurring. Whereas the greater the difference between f(Φ) and f(Sib) the more exploration is being conducted. MMAS appears to be the algorithm of choice here as it has achieved a much fitter solution than BAS. BAS both improves the quality of solutions slower than the GA and provides a less fit final solution and therefore is poorer in both respects. MMAS has achieved a fitter final solution than the GA but requires more fitness evaluations than the GA.

Up to approximately 50,000 fitness evaluations the GA improves the quality of solutions more quickly than MMAS. From 50,000 fitness evaluations onwards the GA sees little to no improvement while MMAS continues to improve the quality of solutions, eventually providing a fitter solution than the GA.

Table 2 shows the average and best final solution fitnesses and produced by BAS and MMAS for Project File 1. Table 1 shows that MMAS has produced the solution with the lowest f. f for the best solution from MMAS is approximately 1,000,000 less than f for the best solution from the GA and 20,000,000 less than the best solution from BAS. On average MMAS provides a fitter final solution than the GA does for setup 1.

Bas does not perform very well on Project File 1 in terms of ability at finding near optimum solutions. BAS prematurely converges at a local maximum on each of the three runs performed. This is attributed to the non-exclusive approach to pheromone addition causing excessive build of pheromone on particular trails.

Table 2 - Final results for all three algorithms on Setup 1

Method	BAS	MMAS	GA
Average	69617655	673481044	
Best	695982293	672193950	673622800
Fitness Evals	81000	234600	100332

The fitness evaluations row in Table 2 displays the amount of fitness evaluations required by each algorithm to find the best solution.

Setup 2

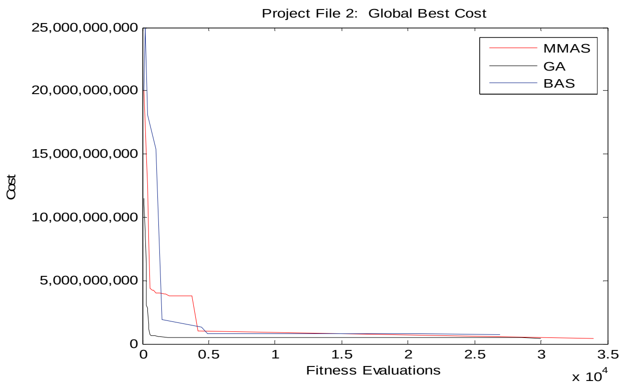

Figure 3 - Fitness of the best solution for BAS, MMAS and the GA using Setup 2

Figure 3 shows a trace of the three algorithms on problem setup 2. As can be seen BAS improves the quality of solutions quicker than MMAS initially but ultimately fails to find a fitter solution than either of the other algroithms. However, BAS sees slower improvement in the fitness of solutions than the GA and results in a less fit final solution. MMAS displays the slowest initial improvement in quality of solutions but proceeds to find the fittest solution overall.

Table 3 shows that MMAS has produced the solution with the lowest fitness but required more fitness evaluations than the GA to achieve this.

Table 3 - Average and best results for all algorithms on Setup 2

	BAS	MMAS	GA
Average	706176556	500905643	
Best	696982293	382092863	448147077
Fitness Evals	25100	34000	30106

Table 3 shows that the solution with the lowest fitness produced by BAS is nearly 300,000,000 higher than the GA. The performance by BAS is significantly worse than the GA and MMAS on these problems and the was made decision not to include BAS in the following experiments due to the consistently poor performance of BAS over Setup 1 and 2.

Accelerating ACO

It has been shown that MMAS is capable of finding solutions with lower fitness than the GA. The focus, therefore, is now on the number of fitness evaluations required. MMAS will be encouraged to converge quicker in the following short experiments by adjusting ρ. Stützle and Hoos (200) showed that a higher ρ value leads to slower convergence and greater chance of finding a near optimum solution therefore, in the following experiments the effect of altering ρ was be examined. It is expected that lowering the value of ρ will initially show quicker improvement in the quality of solutions but will lead to less fit final solutions. These experiments are performed using setup 1 and run for 5000 fitness evaluations.

ERMMAS is included to replace BAS in the following experiments. Figure 4 displays the results of the ρ-value experiments for MMAS and shows that a value of 0.75 for ρ encourages the fastest convergence and a value of 0.99 results in the slowest convergence as expected. These results therefore confirm those found by Stutzle and Hoos (2000).

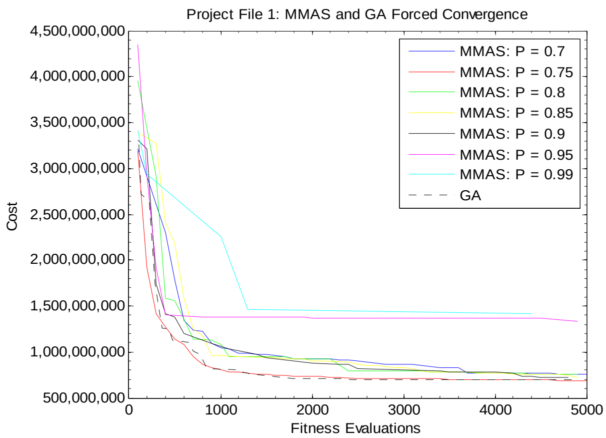

Figure 4 - Comparison of fitness traces for MMAS with varying p-values and the GA for Setup 1.

With ρ = 0.75 MMAS improves the quality of solutions quicker than the GA for the first 400 fitness evaluation and the last 1500 iterations resulting in a fitter solution being found by MMAS. In fact the performance of MMAS with this setting is very close to that of the GA and for many parts of the curve, is better.

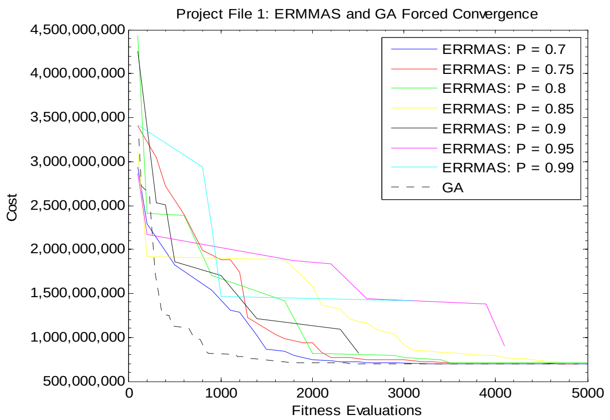

Figure 5 - Comparison of fitness traces for ERMMAS with varying p-values and the GA for Setup 1.

Figure 5 shows that a ρ value of 0.7 has achieved the quickest convergence leading to the fittest solution found by ERMMAS. However, ERMMAS sees significantly slower improvement in the quality of solutions than the GA until around 3500 evaluations. The difference in speed at which the quality of solutions is improved is large when using lower ρ values such as 0.7 or 0.75 compared to ρ = 0.99 for MMAS and ERMMAS.

Table 4 - Comparison of best solution fitnesses all algorithms over 5000 evaluations on Setup 1

P	MMAS	ERMMAS	GA
0.7	712382054	694421933	695426474
0.75	689732784	694555138	
0.8	707236067	703178145	
0.85	704312270	704279302	
0.9	694350970	743975498	
0.95	758040529	906065510	
0.99	1294801273	1058531197	

Table 4 displays Σ and Ω for MMAS and ERMMAS when using problem Setup 1 and a variety a values for the ρ. Table 4 shows that for MMAS a ρ value of 0.9 results in the fittest solution for all the algorithms. Whereas for ERMMAS, a ρ value of 0.7 provided the lowest cost solutions in this shorter timeframe.

4 CONCLUSIONS

In this study we have shown that ACO techniques can be used as effective competitor approaches to genetic algorithms in this important problem domain. The MMAS and ERMMAS ACO algorithms have shown the capability to discover more optimal results that the GA over long-term optimisation runs. This was to a certain extent unexpected as the objective of the study was to show that ACO algorithms could improve on the speed of convergence of the GA, whereas in most cases this wasn't shown. The latter experiments have also shown that the pheromone evaporation rate is crucial to the optimal functioning of the ant system. This is not surprising, but it does highlight the importance of a correctly setup algorithm when dealing with problems of this complexity.

Therefore, while the anticipated computational savings from using ACO did not materialise, some improved results were achieved. There are potentially a variety of reasons why this is the case, including the possibility that the extremely large and rugged search space associated with these problems had a part to play in the slow convergence. It should also be remembered that in the above experiments, the ACO approaches are competing with the best of a number of well-tuned GA runs on the same problem, so the fact that ACO on occasion improves on that result is to its credit. Through this and previous work, it is perhaps becoming apparent that an incremental improvement in WDN optimisation can be achieved through the use of swarm-intelligence based techniques.

An additional advantage to the ACO approach is that it can be more transparent in its decision making that the genetic algorithm. By visualising the pheromone table on the map of possible options, a variety of statistics about the most popular choices for certain components could be generated. Although this has not been investigated in this study, our industrial partners have indicated that this could be a valuable tool in elucidating the decisions being made by the algorithm and therefore increasing confidence in the results.

REFERENCES

[1] Bullnheimer. B., Hartl. R.F., Strauss. C. (1999). A new rank based version of the ant system — a computational study, Central European J. Oper. Res. Econom. 7. pp. 25–38

[2] Beasley. J. E., Chu't'. E C. (1997). A Genetic Algorithm For The Generalised Assignment Problem. Computers Ops Res. Vol. 24, No. 1, Elsevier Science Ltd, pp. 17-23

[3] Dorigo. M., Maniezzo. V., Colorni. A. (1996). The ant system: optimization by a colony of cooperating ants. IEEE Trans. Syst. Man Cybern., 26, pp. 29–42

[4] Dorigo. M., Gambardella. L. (1997). Ant colonies for the traveling salesman problem. BioSystems, 43, pp. 73–81

[5] Dorigo. M., Di Caro. G. (1999). The ant colony optimization metaheuristic. New ideas in optimization. Glover, eds., McGraw-Hill, London, pp. 11–32

[6] Simpson. A. R., Maier. H. R., Foong. W. K., Phang. K. Y. Seah. H. Y., Tan, C. L. (2001). Selection of parameters for ant colony optimization applied to the optimal design of water distribution systems. Proc., Int. Congress on Modelling and Simulation, Canberra, Australia, pp. 1931–1936

[7] Simpson. A. R.., Maier. H. R., Foong. W. K., Phang. K. Y., Seah. H. Y., Tan. C. L. Zecchin. A.C. (2003). Ant Colony Optimization for Design of Water Distribution Systems. Journal of Water Resources Planning and Management, Vol. 129, No. 3, May 1

[8] Stützle. T., Hoos. H.H. (2000). MAX–MIN Ant System. Future Generation Computer Systems 16, Elsevier , pp. 889–914 **able 3:** Forced convergence - Project File 1 varying ρ

[9] Zecchin. A.C., Maier. H.R.., Simpson. A.R., Roberts. A.J. Berrisford. M.J., Leonard. M. (2003). Max-Min Ant System Applied to Water Distribution System Optimisation. In D.A. Post, editor, Proceedings MODSIM 2003: International Congress on Modelling and Simulation, Townsville, Queensland, Australia, July 14–17. pp 795–800

[10] Zecchin. A.C., Maier. H.R.., Simpson. A.R., Leonard. M., Nixon. J.B. (2007). Ant Colony Optimization Applied to Water Distribution System Design: Comparative Study of Five Algorithms. Journal of

Water Resources Planning and Management, Vol. 133, No. 1, January 1
[11] Dandy, G. C., Simpson, A. R., and Murphy, L. J. (1996). An improved genetic algorithm for pipe network optimisation. Water Resour. Res., 32(2), 449–458.
[12] Savic, D. A., and Walters, G. A. (1997). 'Genetic algorithms for the least-cost design of water distribution networks. J. Water Resour. Plng. and Mgmt., ASCE, 123(2), 67–77.
[13] Rossman, L. A. (1999). Computer models/ EPANET. Water Distribution Systems Handbook. Mays, L. W. ed., McGraw-Hill, New York.