Ant Colony Optimisation for the Optimal Control of Pumps in Water Distribution Networks

Manuel López-Ibáñez

PhD Student, CEC, Napier University, Edinburgh, UK

T. Devi Prasad[*]

Lecturer, SEBE, Napier University, Edinburgh, UK

Ben Paechter

Professor, SCS, Napier University, Edinburgh, UK

---

[*] Corresponding Author

Ant Colony Optimisation for the Optimal Control of Pumps in Water Distribution Networks

Manuel López-Ibáñez[1], T. Devi Prasad[2], and Ben Paechter[3]

Abstract: Reducing energy consumption of Water Distribution Networks (WDNs) has never had more significance than nowadays. The greatest energy savings can be obtained by careful scheduling of operations of pumps. Schedules can be defined either implicitly, in terms of other elements of the network such as tank levels, or explicitly by specifying the time during which each pump is on/off. The traditional representation of explicit schedules is a string of binary values with each bit representing pump on/off status during a particular time interval. In this paper a new explicit representation is presented. It is based on time controlled triggers, where the maximum number of pump switches is specified beforehand. In this representation a pump schedule is divided into a series of integers with each integer representing the number of hours for which a pump is active/ inactive. This reduces the number of potential schedules (search space) compared to the binary representation. Ant Colony Optimization (ACO) is a stochastic meta-heuristic for

[1]   Ph.D student, CEC, Napier University, 10 Colinton Road, Edinburgh, UK. M.Lopez-Ibanez@napier.ac.uk

[2]   Lecturer, School of Engineering & the Built Environment, Napier University, 10 Colinton Road, Edinburgh, UK. P.Tumula@napier.ac.uk

[3]   Professor, School of Computer Science, Napier University, 10 Colinton Road, Edinburgh, UK. B.Paechter@napier.ac.uk

combinatorial optimization problems that is inspired by the foraging behavior of some species of ants. In this paper, an application of the ACO framework was developed for the optimal scheduling of pumps. The proposed representation was adapted to an Ant Colony Optimization framework and solved for the optimal pump schedules. Minimization of electrical cost was considered as the objective, while satisfying system constraints. Instead of using a penalty function approach for constraint violations, constraint violations were ordered according to their importance and solutions were ranked based on this order. The proposed approach was tested on a small test network and on a large real-world network. Results are compared with those obtained using a simple Genetic Algorithm based on binary representation and a hybrid Genetic Algorithm that uses level-based triggers.

INTRODUCTION

Operational cost of pumps in a Water Distribution Network (WDN) represents a significant fraction of the total expenditure incurred in the operational management of WDNs worldwide. Pumps consume large amounts of electrical energy for pumping water from source to storage tanks and to demand nodes. They also need to be repaired and replaced, resulting in maintenance costs. Therefore, the goal of a pump scheduling problem is to minimize the total pump operational cost, which includes pumping cost and pump maintenance cost, while guaranteeing a competent network service. In most cases, this problem is equivalent to the minimization of cost of pumping, while supplying water

to consumers at adequate pressures and achieving full recovery of tank levels by the end of operating period.

Finding optimal schedules for pumps in a WDN is a difficult task for researchers and managers alike. It may be due to variability in water consumption, complexity of WDNs and complex electrical tariff structures. Reduction in the total pumping cost in a water supply system can be accomplished in many ways, from proper maintenance of infrastructure to the use of optimal control policies. A careful scheduling of pump operations may shift workload to cheaper electrical tariff periods, and as a result, the cost of energy consumed by pumps is reduced. Furthermore, energy savings can be accomplished by pumping water when tank levels are lower and combining the operations of different pumps efficiently. In general, pump maintenance cost cannot be easily quantified; therefore, surrogate measures are used to estimate it. The most commonly used measure is the total number of pump switches (N_S): frequent switching (on/off) causes wear and tear of pumps, and hence future maintenance costs. These maintenance costs can be limited in the optimization problem using a constraint such as N_S must be less than or equal to a specified value.

Although most of the research in the field of water distribution optimization has been devoted towards the optimal design of WDNs, few researchers have attempted to develop new algorithms for operational optimization of WDNs. Many of these algorithms were developed to determine least-cost pump scheduling policies. These algorithms were developed using linear programming (Jowitt and Germanopoulos, 1992), nonlinear

programming (Chase and Ormsbee, 1993; Yu et al., 1994), dynamic programming (Lansey and Awumah, 1994; Nitivattananon et al., 1996), heuristics (Ormsbee and Reddy, 1995; Leon et al., 2000) and meta-heuristics (Savic et al., 1997; van Zyl et al., 2004; Rao et al., 2005). The difficulties faced by these models in obtaining optimal or near optimal solutions can be attributed to: (i) large search space, and thus, a very limited proportion of solutions can be evaluated in a reasonable length of time; (ii) complexity of real-world water distribution networks, thereby limiting the usefulness of traditional optimization methods such as linear and nonlinear programming; (iii) oversimplification of network model and system of hydraulic equations to suit the algorithm requirements, thereby sacrificing the accuracy; and (iv) tendency of solutions getting trapped at local optima. Algorithms based on meta-heuristics, such as evolutionary algorithms and simulated annealing, have shown promise and are able to overcome some of the above limitations.

One of the drawbacks of population based techniques, such as evolutionary algorithms, is that they require large number of function evaluations before convergence is guaranteed. The search space of the problem grows exponentially with the number of decision variables (i.e., number of control elements such as pumps). This, in turn, increases the number of function evaluations required to adequately explore the search space. Calculation of the objective function value and performance indices of an operational schedule requires conducting an extended period simulation using hydraulic simulator. Therefore, for large complex networks each function evaluation takes up considerable amount of computational time and resources. To avoid the lengthy computational times

researchers have used hybrid methods (van Zyl et al., 2004). Use of hybrid methods requires a user to specify when to switch from global search method to local search method. This switch is highly dependent on the problem under consideration and will be different for different problems.

Operational schedules of pumps in a network can be defined in terms of properties of other elements of the network, called in this study implicit pump scheduling. For example, water levels in tanks are often used to trigger the operation of pumps. Few researches have considered this representation for pump operation (Atkinson et al., 2000; van Zyl et al., 2004). On the other hand, pumps may also be controlled directly by specifying the time during which a pump is on/off, called herein explicit pump scheduling (Savic et al., 1997). Most of the works on explicit pump scheduling have encoded pump schedules using a binary string representing the on/off state of a pump during a pumping interval. In this study, we propose a new representation based on explicit pump scheduling and using time controlled triggers. The new representation is developed by incorporating constraint on the number of pump switches into it. This representation enables the optimization algorithm to conduct search in a reduced search space, which leads to reduction in the number of functional evaluations and, hence, computation time. For example, let us consider a single pump schedule (24 hours) containing on/off states of a pump (binary representation) in one hour intervals. There are 12 possible pump switches and search space contains $2^{24} = 16,777,216$ solutions. However, if the number of pump switches is restricted to three ($N_S \leq 3$), the feasible

search space with respect to this constraint will be reduced to 290,998 solutions, which is less than 1.73% of the total search space. Table 1 gives the number of potential solutions with respect to the number of pump switches. These values are obtained by enumerating a 24 hr pump schedule. As can be observed from Table 1, by incorporating this constraint on number of pump switches into the representation, search space could be reduced considerably and direct the search in the reduced space. This will further lead to reduction in the number of function evaluations required to achieve convergence.

Ant Colony Optimization (Dorigo and Stutzle, 2004) is a biologically inspired optimization method that mimics the technique used by real ants for optimizing the path between nest and source of food. Ants follow chemical substances, called pheromones, previously laid by other ants in their trails. The trail followed by an ant depends strongly on the amount of pheromone present at each possible direction. Thus, pheromones work as a communication mechanism and also as a reinforcement learning process. Ant Colony Optimization has been applied by researchers to the design of Water Distribution Networks (Maier et al., 2003).

The present work focuses on the application of ACO to the problem of operational optimization of WDNs. The pump scheduling problem along with the proposed new representation are adapted to the ACO framework and evaluated by an algorithm similar to the Ant System (Dorigo et al., 1996), which is considered to be the first ACO algorithm. The proposed method is applied to an example network and a real water distribution network. The efficacy of the method is demonstrated by comparing the

results of the present method with those published in the literature. The paper is structured as follows. First, the pump scheduling problem is formally defined. Next, the application of Ant Colony Optimization to this problem is explained. Experimental results are presented and discussed followed by a summary of conclusions and enumeration of future research directions.

PUMP SCHEDULING PROBLEM

Given a water distribution network where demand patterns, initial tank levels and electricity tariffs are specified, the goal is to find the best pump schedule over a typical operating cycle such that the total operational costs are minimized while guaranteeing a competent network service. Pump operational costs include cost of energy consumed by pumps and pump maintenance costs derived from the workload imposed on pumps. System constraints ensure feasibility of pump schedules, including that demands are supplied at adequate pressures and water supplied from tanks is recovered by the end of the scheduling period.

Pump energy costs depend on the energy price as well as on the amount of energy consumed. The price per unit of energy is given by electricity tariff, which may vary during a scheduling period. In general, it is divided into an expensive peak and a cheaper off-peak electricity tariffs. The actual amount of energy consumed by a pump depends on several parameters, including flow through the pump, head supplied by the pump and wire-to-water efficiency. These parameters can be calculated using a hydraulic simulator

for a known pump schedule. Formally, operation of N_p pumps in a WDN is scheduled over a scheduling period T. This scheduling period is divided into a number of time intervals N_T. Given a particular schedule S, representing which pumps operate during which time interval, the total cost of energy is calculated as:

$$C_E(S) = \sum_{n=1}^{N_p} \left(\text{demand charge} + \text{consumption charge} \right)$$
$$= \sum_{n=1}^{N_p} \left[R_d(n) \cdot E_d(n) + \sum_{i=1}^{N_T} R_e(i) \cdot E_c(n,i) \cdot S(n,i) \right] \tag{1}$$

where Np = number of pumps; N_T = number of time intervals; S(n,i) = duration for which pump n is operating during interval i (hour); $R_e(i)$ = energy tariff during interval i (\$/kWh); $E_c(n,i)$ = energy consumption rate of pump n during interval i (kWh/h); $R_d(n)$ = maximum demand charge of pump n (\$/kW); and $E_d(n)$ = maximum energy consumption of pump n (kW).

The energy consumption rate of a pump depends on flow through the pump, head supplied by the pump, and efficiency at which it is operating.

$$E_c(n,i) = \frac{0.01019 \cdot Q(n,i) \cdot h(n,i)}{e(n,i)} \tag{2}$$

where Q(n,i) = flow rate through pump n during interval i (l/s); h(n,i) = total dynamic head (TDH) supplied by pump n during interval i (m); and e(n,i) = overall (wire-to-water) efficiency of pump n during interval i.

Similarly, the maximum energy consumption is calculated as:

$$E_d(n) = \frac{0.0098 \cdot Q^{\max}(n) \cdot h(n)}{e(n)} \qquad\qquad (3)$$

where $Q^{\max}(n)$ = peak flow rate through pump n (l/s); h(n) = total dynamic head (TDH) supplied by pump n (m); e(n) = overall (wire-to-water) efficiency of pump n. Although the network instances considered in our study do not include any demand charge, it can be incorporated by adding its cost to the total energy cost as defined in Eq. (1).

On the other hand, maintenance costs are difficult to quantify and are usually measured using a surrogate objective, such as the number of pump switches. A pump switch is defined as turning a pump on that was off during preceding time interval. Frequent switching of pumps causes wear and tear of pumps, which, in turn, increases maintenance costs. Thus, it is assumed that minimization of the number of pump switches reduces maintenance costs (Lansey and Awumah, 1994). Nonetheless, our approach does not minimize the number of pump switches, rather it is considered as a constraint that should not exceed a specified value. Such constraint in the number of pump switches is implicitly enforced by the representation of schedules presented in this paper.

Constraints

In order to obtain feasible pump schedules, the optimization model must satisfy system constraints that represent its performance criteria. These include hydraulic constraints representing conservation of mass and energy, minimum and maximum limits on tank storage levels, minimum pressures requirements at demand nodes and a balance between supply and demand from tanks. The hydraulic simulator implicitly handles some of these

9

constraints, including hydraulic constraints and constraints on maximum and minimum tank levels.

To achieve a balance between water supplied and consumed from tanks, a viable schedule must ensure that tanks recover their levels by the end of scheduling period. That is, tank levels at the end of the scheduling period are not lower than those at the start. Tank volume deficit (ΔV_k) is defined as the difference in percentage between initial volume ($V_{k,S}$) and final volume ($V_{k,E}$) of water in a tank. A negative volume deficit represents a surplus of water in a tank. However, in this work it is not assumed that this surplus compensates the loss of water in a different tank. Therefore, only positive volume deficits are accumulated to calculate the total volume deficit (ΔV) of a particular schedule. This value must be equal to zero for any feasible solution:

$$\Delta V_k = \begin{cases} \dfrac{V_{k,S} - V_{k,E}}{V_{k,S}} \times 100 & if \ V_{k,S} > V_{k,E} \\ 0 & otherwise \end{cases} \qquad k = 1,\ldots,N_t$$

(4)

$$\Delta V = \sum_{k=1}^{N_t} \Delta V_k = 0$$

(5)

where N_t = number of tanks in the network.

Consumers must be supplied water at adequate pressures. Therefore, the optimisation model must satisfy minimum pressure constraints at demand nodes:

$$H_{j,i} > H_j^{\min} \qquad j = 1,\ldots,N_d$$

(6)

where $H_{j,i}$ = head supplied at demand node j during time period i; $H_j^{\min}$ = is the minimum head required at demand node j; and N_d = number of demand nodes.

In order to reduce maintenance costs, an additional constraint on the number of pump switches is used. Thus, the number of pump switches is limited to a specified value:

$$N_S = N_p \times SW \tag{7}$$

where SW is a constant to be specified and it is the maximum number of switches allowed per pump during a scheduling period. Schedules with a lower number of pump switches are also acceptable, and thus, constraint Eq. 7 may be relaxed as follows:

$$N_S \leq N_p \times SW \tag{8}$$

Finally, the hydraulic simulator may issue *warnings* when some schedules are evaluated. These are the warnings generated by the simulator when it encounters situations such as when a pump cannot supply sufficient head. Whenever pumps cannot deliver enough head or flow - one or more pumps will be forced to either shut down (due to insufficient head) or operate beyond the maximum rated flow (EPANET simulator warning number 4). From efficiency point of view, operating a pump at extreme points on the pump curve is not desirable and, in some cases, an extra pump switch may be introduced due to shutting-off of pump(s). These warnings are not specific to EPANET rather they are due to inefficient operation of pumps. Similarly EPANET will generate warnings if valves cannot deliver enough flow (EPANET Warning 5). In this study we have accumulated the

number of such warnings generated by the simulator and used in the constraint handling procedure. Feasible schedules should not generate simulation warnings.

Constraints Handling Methodology

Hydraulic constraints and limits on tank levels are enforced implicitly by the simulator used in this study. The constraint on the number of pump switches, (i.e., either Eq. 7 or Eq. 8), is implicitly satisfied by the time-based triggers representation. On the other hand, constraints on total volume deficit, Eq. 5, and minimum pressure requirements at demand nodes, Eq. 6, must be explicitly handled by the optimisation algorithm.

Previous studies have dealt with constraints by penalising the objective function (Mackle et al., 1995; 2000; Boulos et al., 2001; and van Zyl et al., 2004). This requires the definition of a penalty function and appropriate penalty values. Moreover, different penalty values are required for different types of constraints and the degree of violation of some of these constraints cannot be easily quantified. Penalty values, in general, are obtained either using rudimentary techniques or by trail-and-error. The penalty function approach imposes a fixed trade-off between the amount of constraint violation and the value of the objective function. Low penalty values would allow constraint violations in return for small reductions in the objective value, while higher penalty values would require higher objective value differences to compensate the same amount of constraint violation. For these reasons, the use of penalty function approach is avoided in this study. The present paper considers an approach based on ranking solutions with respect to their

constraint violations and objective function values. In this sequential approach, given two candidate solutions, the criteria used to choose a better solution are:

1. select the solution with the lowest pressure violation;

2. if pressure violations are equal, select the solution with the lower number of warnings from the simulator;

3. for equal number of warnings, select the solution with the lower total volume deficit (ΔV) ;

4. if total volume deficits are equal, select the solution with the lowest objective function value (C_E).

These criteria effectively rank a feasible solution (zero total volume deficit, no warnings and no pressure violations) better than any infeasible one. Feasible solutions are compared with respect to their objective function values only, and infeasible solutions are compared according to their degree of infeasibility. The order chosen for the comparison of constraint violations establishes some preference for constraint violations. A solution with tank volume deficit, where enough water is supplied to meet the demand but a balance is not achieved at the end of the simulation, is preferred over a solution having pressure violations, where the adequate demand cannot be supplied. Warnings (described above) from the simulator are considered to be worse than tank volume deficit, since warnings indicate inefficient operation of pumps or some other problem with the schedules preventing correct evaluation of the solution by the simulator. However, a

solution with warnings and no pressure violations is preferred over a solution that has pressure violations. It was observed that a small modification to such a solution removes warnings. On the other hand, a solution with pressure violations require more fine-tuning, that is, pumps are required to be active for more hours to supply the demand at required pressures.

ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is a meta-heuristic approach successfully applied to several optimization problems. Dorigo and Stutzle (2004) have provided an extensive introduction to this technique. ACO was inspired by the foraging behavior of some species of real ants. These ants are able to find an optimal path between nest and food through indirect communication known as stigmergy, by means of trails of a chemical substance called pheromone laid by ants along their way when moving between nest and food. ACO algorithm makes use of very simple agents, analogous to artificial ants, which stochastically build paths in a graph. Such graph represents an optimization problem, where nodes are decision points and edges represent possible choices or solution components. A path over the graph defines a candidate solution to the optimization problem (see Fig. 1). At each decision point (node), an individual ant stochastically chooses which solution component (edge), to add to its current path. Solution components are added iteratively until a candidate solution is completed. Each stochastic decision is influenced by numerical information associated with each edge in the graph,

analogous to pheromone trails. The higher the pheromone value of a particular solution component, the greater its probability of selection by an ant to add it to its current partial solution. Ants modify this pheromone information and increase the probability of choosing the most attractive edges of the graph. Additionally, all pheromone values are decreased every iteration, an operation known as evaporation. This process decreases pheromone values associated with solution components that seldom appear in good solutions and that have not been recently reinforced, so the search is directed towards the most promising regions of the search space.

This work adapts the original Ant System (AS) proposed by Dorigo et al. (1996), which was the first ACO algorithm, with some minor modifications. The schema of the resulting algorithm is shown in Fig. 2. Let (i,j) be the jth edge originating from node i and representing solution component (i,j). Let $\tau_{ij}(t)$ be a numerical value associated to each solution component (i,j), called pheromone. This pheromone value is updated during the algorithm run, and it depends on the current iteration t. First, pheromone values are initialized to a constant value τ_0. Then, a main loop is repeated until termination criteria, such as maximum number of objective function evaluations, is met. Within this loop, a set of ants (A) construct candidate solutions to the problem. Each ant 'a' constructs a single solution ϕ_a by iteratively adding solution components, which is equivalent to building a path in the graph by choosing edges at each decision point. Thus, an ant 'a' at each decision point i chooses a single solution component (i,j) from among a set $N_i(a)$ of possible alternatives. This set $N_i(a)$ is called the feasible neighborhood of ant 'a' and it

may vary according to the current partial solution of ant 'a'. The probability of choosing a solution component (i,j) is given by:

$$p_{ij}(a,t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_i(a)}\left[\tau_{il}(t)\right]^{\alpha}\left[\eta_{il}\right]^{\beta}} \tag{9}$$

where η_{ij} is a heuristic value associated to component (i,j), and represents an estimation of the suitability of choosing j over the other alternatives in $N_i(a)$; and α and β weigh the relative influence of pheromone and heuristic information on the final probability $p_{ij}(a, t)$.

After each ant has constructed a new solution, they are evaluated and ranked to identify the iteration-best solution (ϕ^{ib}), the best solution among the ones constructed in the current iteration 't'. Also, the best solution found in the current run of the algorithm, called best-so-far, (ϕ^{bf}), is updated accordingly. Finally, pheromone information is updated to reflect the experience acquired by the ants through the evaluation of their solutions. The pheromone update is completed in two steps. In the first step, pheromones of all solution components are evaporated by decreasing pheromone values by a constant factor. In the second step, pheromones of solution components that are part of the best solutions are reinforced by increasing their pheromone values. Both operations, evaporation and reinforcement, can be formalized as:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \tag{10}$$

where $\rho \in [0, 1]$ is a parameter that determines the evaporation rate $(1-\rho)$ and $\Delta\tau_{ij}$ is the amount of pheromone deposited in solution components belonging to the iteration-best solution (ϕ^{ib}) as given by:

$$\Delta\tau_{ij} = \begin{cases} \Delta\tau & \text{if edge } (i,j) \text{ is part of } \varphi^{ib} \\ 0 & \text{otherwise} \end{cases} \qquad (11)$$

where $\Delta\tau$ is a user-defined constant. According to Dorigo and Stuetzle (2004), in the simplest case $\Delta\tau$ can be the same constant value for all ants. The only requirement is that $\Delta\tau$ is non-increasing value with respect to the objective function value. In this case, only the differential cost works in favor of the detection of least-cost components. From our experiments it was found that $\Delta\tau=1$ produced better results compared to a value based on solution quality.

This pheromone reinforcement method is different from the update method used in the original Ant System (Dorigo et al., 1996), where all ants in the current iteration deposited an amount of pheromone relative to its objective function value. Nevertheless, update methods that use just one solution to reinforce the pheromone values, either the iteration-best or best-so-far solutions, are widely used in many modern ACO algorithms to achieve a faster convergence (Dorigo and Gambardella, 1997; Stutzle and Hoos, 2000; Dorigo and Stutzle, 2004).

REPRESENTATION OF PUMP SCHEDULES

An explicit pump schedule defines time intervals during which each pump in a network is on or off. In an implicit formulation, decision variables are system conditions, such as tank water levels, that trigger the operation of the pumps. However, the drawback with implicit approach is that one has to assign each pump to a particular tank a priori. In some cases the optimal solution needs to be converted to obtain an actual pump schedule, which may not be optimal. In this paper, a new explicit formulation of the pump scheduling problem, where decision variables are on/off durations of each pump, is developed.

An explicit formulation of decision variables can be represented in several ways, binary representation being the most commonly used. In binary representation, the scheduling period (T) is divided into a fixed number (N_T) of smaller intervals and a single bit is used to represent the status of a pump during each interval. For example, a 24 hour scheduling period is divided into 24 one-hour intervals. In this case, the maximum number of switches per pump would be 12, which is much larger than the number of switches that would be allowed in any practical solution. In this study, instead of binary representation for decision variables, a representation based on time-controlled triggers similar to the one proposed by López-Ibáñez et al. (2005) is developed. It has the advantage of implicitly satisfying the constraint on the number of pump switches.

In the time-based triggers representation, each pump switch is defined using a pair of integers. These integers represent the time for which a pump is inactive and active,

respectively, in a pump switch. Therefore, for a pump (n) with SW pump switches there will be (2·SW) decision variables stored in a vector ϕ^n. For a feasible pump schedule, the sum of all integers in each vector ϕ^n must be equal to the scheduling period T. Thus, a candidate solution (S) of the pump scheduling problem has the following form:

$$S = \left\{ \varphi^1, \ldots, \varphi^n, \ldots, \varphi^{N_p} \right\} \quad \text{where each} \quad \varphi^n = \left\{ \varphi_1, \ldots, \varphi_i, \ldots, \varphi_{2 \cdot SW} \right\} \tag{12}$$

such that:

$$\sum_{i=1}^{2 \cdot SW} \varphi_i = T \tag{13}$$

Finally, we have to decide the range of values each ϕ_i can take such that the number of pump switches should not exceed N_S, while satisfying Eq. 13 for each pump. There exist two scenarios. In the first case, to utilize the maximum allowed pump switches, constraint Eq. 7 must be enforced. That is, each integer ϕ_i must be greater than zero. If there are (2·SW−1) integers having a value of 1 hour each, then, following constraint Eq. 13, the remaining integer must be equal to (T−2·SW+1) hours. Therefore, $\phi_i \in [1, (T−2 \cdot SW+1)]$. On the other hand, if Eq. 8 is used then $\phi_i \in [0, T]$. An example of time-based triggers representation for a single pump with SW = 3 and T = 24 hours in shown in Fig. 3.

APPLICATION OF THE MODEL

In order to apply the above defined ACO framework to the Pump Scheduling problem, the time-based triggers representation has to be translated into a graph formulation. For simplicity let us restrict first to the problem of scheduling a single pump with a maximum of SW pump switches. By definition, each pump switch involves a pair of time intervals during which a pump is, respectively, off and on. A complete schedule contains (2·SW) intervals. If the constraint on pump switches as defined by Eq. 7 is considered, then each interval can take any value between 1and (T−2·SW+1). On the other hand, if the more relaxed constraint given by Eq. 8 is used, then each interval can take any value between 0 and T. Therefore, the pump scheduling problem can be adapted to the ACO framework by associating each decision point to an interval and by defining each solution component (i, j) as the assignment of number of hours to an interval.

Since the total duration of all operational intervals must be equal to T, the feasible neighborhood of ant 'a' at decision point i, as defined by Eq. 9, is limited to

$$N_i(a) = \begin{cases} \{1, 2, \ldots, (T - 2 \cdot SW + 1 - T_a)\} & \text{if Eq. 7 is used} \\ \{0, 1, \ldots (T - T_a)\} & \text{if Eq. 8 is used} \end{cases}$$

$$(14)$$

where T_a is the summation of intervals already assigned in the partial solution being constructed by ant 'a'. After one solution component (i,j) has been chosen by ant 'a' to be added to its partial solution. T_a is updated by adding the number of hours assigned to

20

solution component (i,j) to it. The last solution component is not chosen stochastically; on the contrary the last edge is assigned a number of hours such that the total duration of the schedule is T, to satisfy constraint Eq. 13 of the time-based triggers representation. However, this will cause the last interval to have a higher probability of being assigned a shorter duration. To avoid introducing such a bias, each interval is considered in a random order, so that the last assigned interval does not necessarily correspond to the last interval of the schedule. To make linear influence, η_{ij} value was taken as equal to $(T - j)/T$ for time intervals during which a pump is on and otherwise it is equal to (j/T). If either value is smaller than 0.001 then it is taken as 0.001 to allow pheromones to have some influence in the result.

The formulation described above is extended to more than one pump by associating a separate pheromone matrix to each pump. The schedule of each pump is constructed by using the values of its own pheromone matrix. This is equivalent to associating one independent graph to each pump, such that each ant constructs one path in each graph in order to build a complete schedule of all pumps. Similarly, each pheromone matrix is updated following Eq. 11, but only those solution components of ϕ^{ib} that are associated to the pump.

RESULTS AND DISCUSSION

For comparison purposes, the proposed ACO approach was tested on two network instances published in the literature: van Zyl test network and Richmond network. Typical pump scheduling period is 24 hours, whereas billing period is one month. Demand cost, in general, is calculated over a billing period and is not considered (i.e., $R_d = 0$) in the present study. The objective is to minimize C_E as defined by Eq. 1, and subject to constraint Eqs. 5 and 6. The optimization algorithm must handle constraint Eqs. 5 and 6, where as conservation of mass and energy equations are implicitly satisfied by the simulator. A limit of SW = 3 switches per pump is set in all the examples solved using ACO. Thus, the maximum possible value of total number of pump switches (N_S) of any solution is 9 for van Zyl test network and 21 for Richmond network.

For each example, 25 runs were conducted using different random seeds to assess ACO model's average performance. In order to enable a fair comparison with the results provided by van Zyl et al. (2004), each run was continued until the same number of function evaluations, i.e., 6000 function evaluations for van Zyl instance and 8000 function evaluations for Richmond instance. Results obtained are compared with those obtained using simple GA (SGA) with binary representation and HybridGA with level controlled triggers representation.

Van Zyl test network

Initially the test network, published in van Zyl et al. (2004), was solved using the proposed model. It contains 3 pumps and 2 tanks. The layout of the network is shown in Fig. 4. Pumps 1A and 2B are identical pumps connected in parallel. When neither of these pumps is active, a booster pump 3B transfers water from tank A to tank B. In case one or both the pumps (1A and 2B) are active, pump 3B boosts the flow to tank B. Tank B has a higher elevation than tank A, and thus, water may flow by gravity from tank B to tank A through the pipes connected to the demand node. The pump scheduling period T (24 hrs) is divided into twenty-four 1 hr intervals.

Investigations were carried out to fine-tune the ACO parameters. Mainly, the number of ants (A), pheromone update parameter ($\Delta\tau$), and parameters α and β were considered for fine-tuning. As a result of these trials, the following parameter values were found to produce acceptable results and adopted for this example in the pheromone updating process: the number of ants $A = 10$, the evaporation parameter $\rho = 0.95$, the initial pheromone $\tau_0 = 1$, pheromone update parameter $\Delta\tau = 1$, $\alpha = 1$, and $\beta = 0.25$. The median, standard deviation, best and worst values were obtained from the results of 25 runs and presented in Table 2. C_E and N_S correspond, respectively, to the daily electrical cost and total number of pump switches. Results under ACO-7 were obtained taking into account constraint Eq. 7, while results under ACO-8 were obtained considering constraint Eq. 8. In Table 2, results obtained using HybridGA (van Zyl et al., 2004) are also presented. The ACO algorithm produced much better results compared to minimum cost solution and

compare well with the median results obtained using HybridGA. The statistical parameters of HybridGA, presented in Table 2, were calculated based on results from 7 random runs. It should be noted that, in case of ACO, no local search method was employed and a simple ACO algorithm was used. Unlike the Hybrid GA, it does not require fine tuning of the penalty cost.

The search space of ACO-7, for this problem, contains 1.95×10^{16} potential solutions whereas the search space of ACO-8 contains 2.46×10^{16} potential solutions. Comparing the median electrical costs and number of pump switches obtained by ACO-7 and ACO-8, the results show that by relaxing the constraint on the number of pump switches, these can be notably reduced. Although, HybridGA obtained slightly lower median value, the small difference may be due to the number of solutions utilized in calculating statistical parameters and the stochastic nature of both the algorithms. To ascertain this aspect, a statistical test was conducted to assess whether there was any significant difference between objective values obtained by ACO-7 and HybridGA. A non-parametric statistical test, which does not require the data to be normally distributed, was used for this purpose. The standard non-parametric statistical test used was Wilcoxon rank sum test, also called Mann-Whitney U test (Furlong et al., 2000). The result of the test has shown that there was no statistically significant difference between the median electrical costs obtained by ACO-7 and HybridGA (at a critical value of 0.05). Similar statistical results were obtained for ACO-8 also.

Richmond test network

To test the efficacy of the proposed ACO model, subsequently, it was applied to the Richmond water distribution system, which is a real system located in the United Kingdom. The calibrated network has 7 pumps, 6 tanks and a reservoir. In total, the network consists of 948 links and 836 nodes. Complete description of the Richmond network can be found in van Zyl et al. (2004). This network was first studied by Atkinson et al. (2000) and later by van Zyl et al. (2004). The network used in this study is the same as that of van Zyl et al. (2004). All tanks in the network must be 95% full at the start of the peak electricity period (7:00 am).

Some initial trials were carried out to find optimal ACO parameters. As a result of these trials, the following parameter values were found to produce better results and adopted for this example: the number of ants $A = 10$, the evaporation parameter $\rho = 0.95$, the initial pheromone $\tau_0 = 1$, pheromone update parameter $\Delta\tau = 1$, $\alpha = 1$, and $\beta = 0.25$. EPANET hydraulic simulator (Rossman, 1999) was used for calculating objective function values and constraint violations. Although constraints on minimum pressure requirements and tank volume deficits were explicitly handled in the optimisation, EPANET generated warnings for some of these solutions. These warnings, as explained earlier, were related to pumps that cannot deliver sufficient head. Therefore, in this study all such warnings were counted and used in the constraint handling procedure. Two different case scenarios, explained in the following paragraphs, were investigated.

25

Case 1

In this case, the Richmond network was solved considering 1 hr time step, i.e., pumping schedule (T) was divided into 24 equal intervals. Twenty five runs were conducted using the values of ACO parameters described above and different random seeds. Each run using ACO was continued until 50,000 function evaluations and results at the end of 8000 and 50,000 function evaluations were used for comparison purposes. The median, standard deviation, best and worst objective function values were calculated from the results of 25 runs and presented in Table 3. The results under heading ACO-7 were obtained by solving ACO with Eq. 7, and results under heading ACO-8 were obtained by solving ACO with Eq. 8. It can be observed from Table 3 that the best solution of ACO-7 (after 8000 function evaluations) has an objective function value of 33,683.3 and that of ACO-8 has an objective function value of 34,319.9. Slightly larger objective function value could be attributed to the larger search space of ACO-8. For this problem, the search space of ACO-7 contains 1.1×10^{38} potential solutions, whereas ACO-8 contains 1.8×10^{38} potential solutions. Also, from Eq. 14 it can be deduced that the search space of ACO-7 is a subset of the search space of ACO-8. This indicates that the stopping criteria adopted has an effect on the convergence of the algorithm. To investigate this aspect further, results at the end of 50,000 function evaluations were analyzed. It can be observed from Table 3 that ACO-8 was able to identify better solutions compared to ACO-7, when continued to run for longer number of evaluations.

26

For comparison purposes, the Richmond network was also solved using a simple Genetic Algorithm (SGA). In this, each pump schedule was encoded as a binary string with each bit representing on/off status of a pump during one hour time interval. This representation will produce pump schedules similar to those obtained by the ACO model. Unlike the ACO model, the SGA model does not incorporate a constraint on the number of pump switches into the representation. Therefore, the maximum number of pump switches per pump using this representation is 12 (one hour pumping intervals). To restrict the number of pump switches per pump to a maximum of four, constraint Eq. 8 was used. This constraint along with the other constraints, were handled using the constraint handling methodology described above. The SGA was run 10 times with different random seeds and GA parameters. The fine-tuned SGA parameters used are: population size = 100 and number of generations = 1000. That is, a total of 100,000 function evaluations were performed in each SGA run. The genetic operators used were binary tournament selection with elitism (the best solution is not replaced), uniform crossover with a probability varying between 0.75 and 1.0, and flip mutation with a probability varying between 0.01 and 0.05. Results are presented in Table 3.

From Table 3, it can be observed that the median SGA solution requires an operational cost of 41,099.0 per annum. This solution was obtained within 100,000 function evaluations. Compared to this, the median solution obtained by ACO-8 requires an annual operating cost of 36,699.7 and this solution was obtained within 8000 function evaluations. The median number of pump switches of ACO-8 is less than a half of that

27

corresponding to SGA, that is, the median schedule of ACO-8 requires 12 pump switches while the one of SGA requires 24.5 pump switches. In conclusion, ACO is able to obtain schedules with lower maintenance and electrical costs than SGA, in a shorter period of computation time.

Case 2

The sensitivity of the ACO algorithm to the coarseness of the time discretization was investigated in this case. The Richmond network was solved by discretizing the pump scheduling period (24 hrs) into equal intervals having time steps of 1 hr., 30 min. and 15 min. That is, the problem was solved by taking $N_T=24$, $N_T=48$ and $N_T=96$. For each N_T value, the ACO-8 was solved 25 times using the same ACO parameters described in Case 1 above and different random seeds. Each run was continued until 8000 function evaluations were completed. Statistical parameters calculated from these results are presented in Table 4. It can be observed from Table 4 that $N_T=24$ produced best results and the results using different N_T values are differing by no more than 3%. Naturally, smaller time steps are expected to produce better results; but, the increased search space requires larger number of function evaluation. As can be observed from Table 4, the search space of the problem with $N_T=24$ contains 1.8×10^{38} potential solutions compared to 2.1×10^{49} potential solutions when $N_T=48$. By running ACO with smaller time steps and for larger number of function evaluations (Case 1 and Table 2), better results could be obtained. It should be noted that the problem was solved using $N_T=96$ (i.e., time step=15 min.) out of academic interest. In the real networks implementation of pump schedules

with pumps off and on in 15 min intervals may not be desirable due to increased pump maintenance cost and damage that may be caused to infrastructure due to frequent switching of pumps.

For comparison purposes the statistical parameters calculated from the results of Hybrid GA (Zyl et al. 2004) are also presented in Table 4. The best results obtained using ACO-8 with different N_T values (Table 4) is better than the best result of HybridGA. It is worth mentioning that van Zyl et al. (2004) did not provide numerical results but objective function values were presented graphically. These graphical results were converted to the best of our ability into numerical values and statistical parameters were calculated (Table 4). Also, results from only 10 HybridGA runs were used in the calculation of these statistical parameters.

The Wilcoxon rank sum test does not find a statistically significant difference in the median electrical costs obtained by ACO-8 (T=24) and Hybrid GA. This result suggests that any perceived difference may be due to the random initial seed rather than a real advantage of one algorithm with respect to the other. However, nothing can be said with regard to maintenance costs, since results provided by van Zyl et al. (2004) do not mention the number of pump switches obtained by Hybrid GA in the Richmond network. The Hybrid GA proposed by van Zyl et al. (2004) generates more flexible schedules since pumps may be triggered at any moment of the day, while in the ACO approach the status of a pump can only change at fixed intervals. In addition, van Zyl et al. (2004) make use of an implicit representation of schedules based on tank-level triggers, which prevents

tanks from becoming empty, thus reducing the risk of pressure violations at demand nodes. In contrast, the time-based triggers representation may generate infeasible solutions with respect to pressure constraints. Finally, the GA proposed by van Zyl et al. (2004) incorporates local search procedures, which they show to be a key feature of their method. Nonetheless, local search methods may be also incorporated to the ACO approach and we expect that local search will enhance the results presented in this paper.

On the other hand, there are some benefits in the ACO algorithm discussed here, which are not present in the approach followed by van Zyl et al. (2004). The flexible schedules generated by the HybridGA may produce extremely short time intervals between successive pump switches, that is, a pump may be turned off and after few minutes turned on again. Apart from the wear and tear that this sudden switching inflicts to pumps, the pressure changes may damage the network, thus increasing maintenance costs. Also, the level-based triggers representation used by van Zyl et al. (2004) may generate a high number of pump switches. Although van Zyl et al. (2004) introduce penalties on the number of pump switches, there is no account on how effective the penalties were in the case of the Richmond network and how many trial-and-error runs were required to find adequate penalty values. No penalties are needed when using time-based triggers, as proposed in this work, and the number of pump switches is limited by a user-defined parameter.

The computational effort required by the ACO approach was measured when run on a Pentium 4 (3.20GHz) with 1024KB of cache size and running under Red Hat GNU/Linux

with kernel version 2.4.20. The mean computation time required for 6000 evaluations of the van Zyl network was 417.6 seconds, while it was 8862 seconds in the case of 8000 evaluations of the Richmond network. However, less than 1 second was consumed by the ACO algorithm (excluding the computation effort spent by the simulator) in the evaluation of the solutions. Thus, the computational overhead of the ACO algorithm is negligible and becomes even less important as the network instance grows in complexity. This result suggests that the highest computation time gains will be obtained by reducing either the number of function evaluations or the time required by each evaluation.

CONCLUSIONS

The present paper makes two significant contributions to the study of optimisation of pump operations in Water Distribution Networks. First, a new representation is proposed based on time-based triggers that turn pumps on/off at certain moments of the scheduling period. Second, the Ant Colony Optimisation meta-heuristic is adapted to solve this optimisation problem.

The advantage of using the time-based triggers representation over traditional binary representation is that constraint on the number of switches per pump can be satisfied implicitly. That is, optimised schedules will have a predefined maximum number of pump switches, thus limiting the wear and tear of the pumps, and limiting maintenance costs. As a consequence of this, the search space, that is, the number of potential solutions can be significantly reduced. Experimental results indicate that schedules

31

obtained using new representation contain equal or lower number of pump switches than schedules obtained by using a binary representation and an explicit constraint on the number of pump switches.

An application of the ACO framework to the optimisation of pump schedules is presented. The optimisation algorithm, based on the Ant System, minimizes the electricity cost of pumps while satisfying constraints on minimum pressures and balance between supply and demand from tanks. In contrast to the previous approaches, the proposed algorithm does not use a penalty function to handle constraint violations. Thus, it does not require definition and fine-tuning of penalty values. Instead, constraint violations are ordered with respect to their perceived importance. This order is used to rank solutions in such a way that feasible schedules are always ranked better than infeasible schedules.

Results of a small test network and a real network show that the performance of the proposed model with a simple ACO algorithm is better than HybridGA. In the case of a large real-world network, the proposed ACO algorithm outperforms a simple Genetic Algorithm in terms of electrical cost, number of pump switches and computation time. The particular ACO algorithm used in this study, the Ant System, is rather simple. The proposed approach can be straightforwardly extended with more complex and modern ACO algorithms, such as Ant Colony System (Dorigo and Gambardella, 1997) and Max-Min Ant System (Stutzle and Hoos, 2000). Another interesting finding is that, for this

problem, the computational effort overhead of the operation of ACO algorithm is very small in comparison to the spent on hydraulic simulations.

REFERENCES

Atkinson, R., van Zyl, J. E., Walters, G. A., and Savic, D. A. (2000). "Genetic algorithm optimisation of level-controlled pumping station operation." *Proc., Water network modelling for optimal design and management*, Centre for Water Systems, Exeter, U.K., 79–90.

Boulos, P. F., Orr, C. H., De Schaetzen, W., Chatila, J. G., Moore, M., Hsiung, P., and Thomas, D. (2001). "Optimal pump operation of water distribution systems using genetic algorithms." *Proc., Distribution System Symp.*, AWWA, San Diego, USA, 23-25.

Chase, D. V. and Ormsbee, L. E. (1993). "Computer generated pumping schedules for satisfying operation objectives." *J. Am. Water Works Assoc.*, 85(7), 54–61.

Dorigo, M. and Gambardella, L. (1997). "Ant Colony System: A cooperative learning approach to the travelling salesman problem." *IEEE Transactions on Evolutionary Computation*, 1(1), 53–66.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). "Ant System: Optimization by a colony of cooperating agents." *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1), 29–41.

Dorigo, M. and Stutzle, T. (2004). *Ant Colony Optimization*, MIT Press, London.

Furlong, N. E., Lovelace, E. A., and Lovelace, K. L. (2000). *Research Methods and Statistics: An Integrated Approach*, Harcourt College Publishers.

Hollander, M. and Wolfe, D. A. (1999). *Nonparametric statistical inference,* 2nd Ed., John Wiley & Sons, New York.

Jowitt, P. W. and Germanopoulos, G. (1992). "Optimal pump scheduling in water supply networks." *J. Water Resour. Plan. Manage.*, ASCE, 118(4), 406–422.

Lansey, K. E. and Awumah, K. (1994). "Optimal pump operations considering pump switches." *J. Water Resour. Plan. Manage.*, ASCE, 120(1), 17–35.

Leon, C., Martin, S., Elena, J. M., and Luque, J. (2000). "EXPLORE: Hybrid expert system for water networks management." *J. Water Resour. Plan. Manage.*, ASCE, 126(2), 65–74.

López-Ibáñez, M., Prasad, T. D., and Paechter, B. (2005). "Optimal pump scheduling: Representation and multiple objectives." *Proc., Eighth International Conference on Computing and Control for the Water Industry*, D. A. Savic, G. A. Walters, R. King, and S. T. Khu, eds., University of Exeter, UK, 117–122.

Mackle, G., Savic, D. A., and Walters, G. A. (1995). "Application of genetic algorithms to pump scheduling for water supply." *Proc., Genetic Algorithms in Engineering Systems: Innovations and Applications*, GALESIA'95, Sheffield, UK, 400–405.

Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y., and Tan, C. L. (2003). "Ant colony optimization for design of water distribution systems." *J. Water Resour. Plan. Manage.*, ASCE, 129(3), 200–209.

Nitivattananon, V., Sadowski, E. C., and Quimpo, R. G. (1996). "Optimization of water supply system operation." *J. Water Resour. Plan. Manage.*, ASCE, 122(5), 374–384.

Ormsbee, L. E. and Reddy, S. L. (1995). "Nonlinear heuristic for pump operations." *J. Water Resour. Plan. Manage.*, ASCE, 121(4), 302–309.

Rao, Z., Wicks, J., and West, S. (2005). "ENCOMS - an energy cost minimization system for real-time, operational control of water distribution networks." *Proc., Eighth International Conference on Computing and Control for the Water Industry*, D. A. Savic, G. A. Walters, R. King, and S. T. Khu, eds., University of Exeter, UK, 85–90.

Rossman, L. A. (2000). EPANET2 user manuel, US Environmental Protection Agency, Cincinnati, Ohio.

Savic, D. A., Walters, G. A., and Schwab, M. (1997). "Multiobjective genetic algorithms for pump scheduling in water supply." *Proc., Evolutionary Computing Workshop*, D. Corne and J. L. Shapiro, eds., Lecture Notes in Computer Science, Springer-Verlag Berlin, 227–236.

Stutzle, T. and Hoos, H. H. (2000). "MAX–MIN Ant System." *Future Generation Computer Systems*, 16(8), 889–914.

van Zyl, J. E., Savic, D. A., and Walters, G. A. (2004). "Operational optimization of water distribution systems using a hybrid genetic algorithm." *J. Water Resour. Plan. Manage.*, ASCE, 130(2), 160–170.

Yu, G., Powell, R. S., and Sterling, M. J. H. (1994). "Optimized pump scheduling in water distribution systems." *J. Optim. Theory Appl.*, 83(3), 463–488.

List of Figures

Figure 1. Representation of a solution as a path in a graph

```
 1: t = 0, initialise pheromones: τ_ij(0) = τ_0
 2: while termination criteria not met do
 3:    for each ant a ∈ {1, ..., A} do
 4:       φ_a = ∅ (empty solution)
 5:       repeat
 6:          calculate probability p_il(a, t) for each l ∈ N_i(a) following Eq. (9)
 7:          stochastically choose solution component (i, j) such as j ∈ N_i(a)
 8:          add solution component to partial solution:   φ_a = φ_a ∪ (i, j)
 9:       until  φ_a is a complete solution
10:    end for
11:    for each ant a ∈ {1, ..., A} do
12:       evaluate solution φ_a
13:    end for
14:    identify iteration-best solution:  φ^iter-best ∈ {φ_1, ..., φ_A}
15:    if φ^iter-best better than φ^best-so-far then
16:       φ^best-so-far = φ^iter-best
17:    end if
18:    evaporate and update pheromone: (Eq. 10)
19:    next iteration:   t = t + 1
20: end while
21: return  φ^best-so-far
```

Figure 2. Algorithmic schema of ACO.
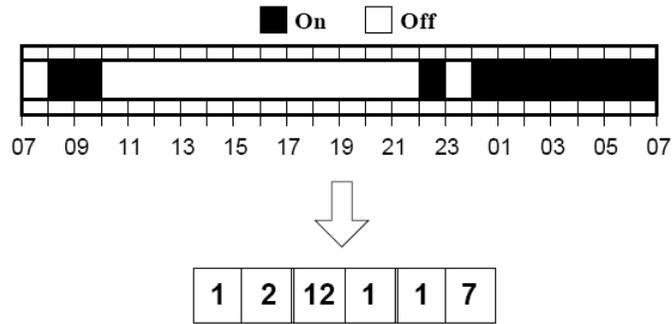
Figure 3. Example of time-based triggers representation
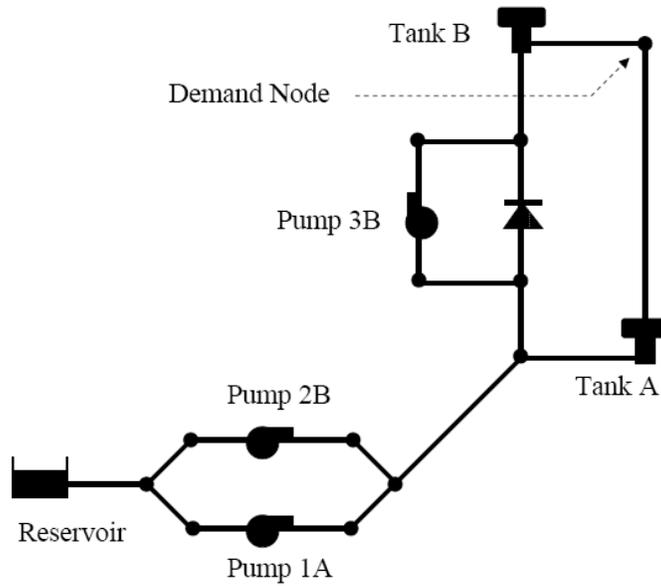
Figure 4. van Zyl test network

Table 1. Reduced space for various Ns values

Ns (≤)	Feasible space with respect to Ns	% of total search space
1	554	0.0033
2	21806	0.13
3	290998	1.73
4	1761940	10.50
5	5684452	33.88
6	11092764	66.11
12	16777216	100.00

Table 2. Results of test network

	ACO-7		ACO-8		Hybrid GA[†]	
	C_E	N_S	C_E	N_S	C_E	N_S
best	329.1	9	326.5	3	344.4	3
median	357.9	9	349.2	4	348.6	4
worst	364.7	9	357.6	7	354.8	5
std. dev.	11.5	0	9.5	1	4.3	1
search space	1.95×10^{16}		2.46×10^{16}		---	

[†]Statistical parameters were obtained based on results from seven HybridGA runs.

Table 3. Results of Richmond network with T = 24.

	ACO		ACO		SGA[†]
	(8,000 evaluations)		(50,000 evaluations)		(100,000
	ACO–7	ACO–8	ACO–7	ACO–8	evaluations)
best	33683.3	34319.9	32728.5	32581.4	37266.5
median	36069.7	36699.7	33747.5	33634.1	41099.0
worst	38017.3	39052.4	35745.5	34464.8	43508.0
std dev.	1075.2	1117.9	544.4	575.2	2291.3
search space	1.1×10^{38}	1.8×10^{38}	1.1×10^{38}	1.8×10^{38}	3.7×10^{50}

[†]Statistical parameters were obtained based on results from 10 SGA runs.

Table 4. Results of Richmond network with different T values.

	ACO–8			HybridGA[†]
	(8,000 evaluations)			(8,000 evaluations)
	$N_T = 24$	$N_T = 48$	$N_T = 96$	
best	34319.9	34848.0	35091.1	35405.0
median	36699.7	37661.1	37898.3	36317.5
worst	39052.4	39545.2	40204.4	37960.0
std. dev.	1117.9	1290.0	1321.8	810.7
search space	1.8×10^{38}	5.9×10^{51}	3.9×10^{64}	----

[†]Statistical parameters were obtained based on results from 10 HybridGA runs.