Journal: JORS Disk used OP: Swarna Ed: Pouly
Article : ppl_jors_2601718 Pages: 1–9 Despatch Date: 7/1/2004

Journal of the Operational Research Society (2003) 0, 1–9 © 2003 Operational Research Society Ltd. All rights reserved. 0160-5682/03 $25.00

www.palgrave-journals.com/jors

Derivation of near-optimal pump schedules for water distribution by simulated annealing

G McCormick* and RS Powell

Brunel University, Middlesex, UK

The scheduling of pumps for clean water distribution is a partially discrete non-linear problem with many variables. The scheduling method described in this paper typically produces costs within 1% of a linear program-based solution, and can incorporate realistic non-linear costs that may be hard to incorporate in linear programming formulations. These costs include pump switching and maximum demand charges. A simplified model is derived from a standard hydraulic simulator. An initial schedule is produced by a descent method. Two-stage simulated annealing then produces solutions in a few minutes. Iterative recalibration ensures that the solution agrees closely with the results from a full hydraulic simulation.

Journal of the Operational Research Society (2004) **0,** 000–000. doi:10.1057/palgrave.jors.2601718

Keywords: water; scheduling; optimization; simulated annealing

Introduction

The problem

After filtration and sterilization, clean water is typically pumped to covered, sterile service reservoirs, from which it gravitates to customers. The use of service reservoirs decouples pumping from demand, which follows a peaky diurnal profile, and creates an opportunity to reduce costs by pumping preferentially at times when electricity tariffs are low, subject to the need to keep enough water in each reservoir for system security. About 15 million tonnes of clean water are pumped each day in England and Wales (http://www.ofwat.gov.uk/pdffiles/leakage.pdf, Table 4), and the power costs of the industry amount to about £110 million per year.[1]

Most pumps are fixed-speed devices, with only on and off settings. Variable-speed pumps are rarer. There may be a few discrete settings to choose, or the small range of available speeds may be approximated by a small number of discrete settings. A pump schedule is thus a valid series of discrete pump settings and switching times. The efficiency of a pump is a nonlinear function of flow that should be near a maximum at its design flow, but reduces for both higher and lower flows. The flow delivered by a pump is an approximately quadratic, decreasing function of the pressure increase across the pump. Network characteristics are also nonlinear: a typical relationship would be $\Delta h \propto q^{1.85}$ where q is the flow and Δh is the 'head' gradient along a pipe (head = pressure plus height). Demands vary continuously throughout the day and it is normally assumed that they do not depend on pressure. Hydraulic simulation requires a series of static solutions of these nonlinear flow and head equations and integration of their effects on reservoir levels. Hydraulic simulation is now a mature technique, and many packages are available (eg Epanet, Ginas, KYPIPE, Stoner, Watnet). Given a validated and calibrated network model with good demand estimates, it is easy to predict the effect of a given pump schedule, even when there are many pump stations and several reservoirs. The inverse problem (produce a cost-effective pump schedule) is rather harder. This is an optimization problem, which could typically be described as:

minimize energy costs while

- keeping source output rates between upper and lower limits;
- keeping reservoir levels in an acceptable range;
- ensuring that reservoir levels at the end of the day are appropriate for the beginning of the next day;
- operating pumps safely (low-efficiency operation causes damage).

There may be other constraints, such as flows not exceeding permitted abstraction limits, upper bounds on pressure in the network, and maximum rates of change of flow through treatment works. Water quality limits may require constraints on the proportions of flow from different sources or restrictions on water travel time to the consumer. Additional costs might include wear and tear on pumps when switching on and charges for maximum electricity demand. The presence and importance of such factors varies a great deal from case to case.

**Correspondence: G McCormick, Water Operational Research Centre, Brunel University, Uxbridge, Middlesex UB8 3PH, UK.*
E-mail: gregory.mccormick@brunel.ac.uk

A practical consideration is that schedules must be produced rapidly if they are to be used operationally—calculation should take substantially less than 15 min for on-line systems, and at most 30 min for 'open-loop' daily calculations.

Previous work

A very wide range of methods has been applied to the pump scheduling problem. The earliest efforts[2] were based on dynamic programming that can cope easily with nonlinearities. Unfortunately, the number of states to consider increases exponentially with the number of reservoirs, which makes this technique impractical when there are more than about three reservoirs in network. Nonlinear programming has been used to solve multiple reservoir scheduling problems.[3,4] It is typically used to calculate optimum reservoir profiles, after which a second stage of calculation is used to find good discrete schedules that achieve the profiles. Since the problem is non-convex, there is no certainty that the global optimum will be found, but good results are reported. Linear programming (LP) is arguably a more flexible approach and has often been used for pump scheduling.[5,6] Naturally, LP depends on finding a suitable linearization of the problem. In some cases, the day is divided into time-slices and the decision variable is the proportion of each time-slice for which each pump is switched on. This still leaves the order of on–off periods to be determined. Some orderings may lead to unacceptable reservoir levels within the time-slice and a collection of heuristics may be needed to find a viable ordering.[6] Mixed integer programming is, in principle, capable of finding a discrete pump schedule directly. Unfortunately, the problem size is often impractical: if the day is divided into only 24 discrete periods then for 35 pumps, there will be at least 840 integer variables. There may be far more variables if the day is divided into more time periods; more accurate models may use 96 periods of 15 min each. A much fuller review of pump scheduling methods will be found in Ormsbee and Lansey.[7] Pump schedule optimization has sometimes met resistance from engineers and controllers because they feel that solutions do not adequately respect subjective or hard to measure considerations that human operators take into account.

Meta-heuristic techniques

Meta-heuristic search techniques such as genetic algorithms and simulated annealing (SA) have been used successfully to solve hydraulic network design problems.[8,9] Applications to pump scheduling are rarer, because of the need to produce solutions rapidly and reliably. Among the few accounts in the literature are Mackle *et al*[10] who applied genetic algorithms to the scheduling of a system with one reservoir and three pumps, and Goldman and Mays[11] who used the same approach on a similar system with water quality constraints. SA operating directly on a hydraulic simulation would be far too slow for routine use. However, if there is a hydraulic linearization that makes LP a viable part of the solution process, that same linearization can speed up SA. Moreover, advantage can be taken of good starting points to further speed up optimization, using two-stage SA.[12] For straightforward problems, this method offers no advantage over LP. However, SA based on linearized hydraulics can potentially cope with nonlinear constraints, nonlinear cost functions, and unique local considerations. This may make solutions more acceptable to network operators and controllers.

This paper describes a hydraulic network linearization based on automatic interaction with Epanet,[13] a hydraulic simulator. A two-stage SA algorithm is then outlined, describing the neighbourhood structure and cooling schedule determination. A previous schedule and/or simple descent may provide a good starting point. SA results are compared with a lower bound from the LP relaxation, and with a progressive mixed integer method. The use of nonlinear cost functions is then discussed. Finally, conclusions are summarized.

Hydraulic network linearization

We have already shown that hydraulic network simulation requires the solution of numbers of simultaneous nonlinear equations. Different scheduling methods have used both implicit variables such as reservoir levels and explicit variables such as pumping durations. We now propose a choice of variables that relate pump scheduling decisions or inputs to scheduling outputs or constraints almost linearly.

Demand variations, and their effects on network flows, linear or otherwise, may be accounted for by dividing the scheduling period into a number of discrete time-slices, during which tariffs are constant and demands are almost constant. We will also assume that all pumps are either off or on throughout a time-slice, so that the resulting pressures flows and efficiencies are constant.

When pumps do not interact hydraulically, for instance when they connect different sources to different reservoirs, their effects on sources and reservoirs can be considered independently. However, when pumps are hydraulically close, there can be significant interactions. For instance, if two pumps force water into the same main, and one generates much higher pressures, the other pump might stall or run inefficiently. Suppose pumps are placed in groups such that pumps that interact with each other are in the same group. Each possible combination of switched-on pumps from such a group may then have unique effects at a given time. By making the decision variable the combination chosen, that is, which combination of pumps will be switched on, and calibrating the effect of each combination, the nonlinear interactions are accounted for. By definition, pumps in different groups will not interact, and the the effect of combinations from different groups will be the linear sum of the individual combinations' effects.

Pressures and therefore flows are affected by changes in reservoir levels. In many networks this effect is small, of the order of 1%. The largest portion of this effect can be taken into account by evaluating linear coefficients assuming a set of typical reservoir profiles. The inaccuracy is also reduced by keeping the time-slices relatively brief.

The linearization described above has similarities with techniques that have been used previously, for example, by Burnell *et al*,[6] and Ulanicki and Orr.[14] The linearized model can be extended to include the first-order effects of reservoir levels on flows, at the cost of some reduction in computational speed. This extension would transform an LP into a quadratic model, but is easy to apply with SA. This extension is not discussed here. In practice, it is also possible to optimize, recalibrate to take account of changes from 'typical' reservoir profiles, and then reoptimize, and in many cases a stable and consistent result can be obtained in this way.

The linearized models used in this paper were based on 24 time-slices of 1 h each. They were built using an automatic process that interacted with the Epanet hydraulic simulator:

1. Identify and remove from consideration closed-loop controls and affected reservoirs.
2. Simulate operation of all individual pumps and all pairs of pumps to identify interactions.
3. Form pumps that may interact nonlinearly into groups.
4. Form all possible combinations of switched-on pumps for each group, including the null combination (all pumps switched off).
5. Remove un-needed combinations (due to identical pumps).
6. Remove combinations that break key constraints (pressure, efficiency, source flow).
7. Calibrate the model by simulating all pump combinations, with the pumps in a combination switched on and all others switched off.

For simplicity, only fixed-speed pump settings (ie on/off) were considered—variable-speed pumps can be dealt with as multiple pumps.

Formulation of the optimization model (linear costs and constraints)

A schedule is a valid assignment of a combination of switched-on pumps from each group in each time-slice. Suppose that the set of valid pump combinations for pump group g is V_g. Let $x_{gc}(t) = 1$ if combination $c \in V_g$ is assigned to group g in time-slice t, otherwise $x_{gc}(t) = 0$. One and only one combination must be chosen for each group and time-slice.

$$\sum_{c \in V_g} x_{gc}(t) = 1 \quad \forall g, t \tag{1}$$

Let r = reservoir, s = source, D = time-slice duration and $\varepsilon_{gc}(t)$ = the energy cost per unit time of combination $c \in V_g$ at time t. The objective function is

Minimize cost = energy cost + penalty costs

$$\begin{aligned} C = & D \sum_g \sum_{c \in V_g} \sum_t \varepsilon_{gc}(t) x_{gc}(t) + \sum_r K_r P_K \\ & + \sum_s \sum_t [U_s(t) p_U + H_s(t) P_H] \\ & + \sum_s Q_s P_Q + \sum_r \sum_t [M_r(t) P_M + A_r(t) P_A O \\ & + B_r(t) P_B + O_r(t) P] \end{aligned} \tag{2}$$

If source costs vary, then charges for water input can also be included.

All costs after the first term are penalty costs, and P_U, P_H, P_Q, P_M, P_A, P_B and P_O are penalty cost coefficients associated with soft constraints. Constraints and associated variables are explained below.

Let L = reservoir level, $\rho_{rgc}(t)$ = a reservoir impact (rate of level change) for the combination $c \in Vg$ at time t. Then for all r, t material balance requires

$$\begin{aligned} L_r(t) = & L_r(t-1) \\ & + D \sum_g \sum_{c \in V_g} \rho_{rgc}(t) x_{gc}(t) \\ & + A_r(t) + B_r(t) \end{aligned} \tag{3}$$

Initial levels $L_r(0)$ are given. Variables A = unmet demand (reservoir level zero) and B = spillage are introduced because of physical limits on reservoir volumes:

$$\forall r, 0 \leqslant L_r(t) \leqslant Lfull_r \tag{4}$$

For security reasons, reservoirs are not normally allowed to empty or fill completely. If M, O, and K represent deviations from specified minimum, maximum and final target reservoir levels, then for all r, t

$$M_r(t) = \mathrm{Max}(Lmin_r - L_r(t), 0) \tag{5}$$

$$O_r(t) = \mathrm{Max}(L_r(t) Lmax_r, 0) \tag{6}$$

$$K_r = \mathrm{Max}(Ltarget_r - L_r(t = last), 0) \tag{7}$$

Source flows F are calculated from the schedule and the linear model.

$$F_s(t) = \sum_g \sum_{c \in V_g} \phi_{sgc}(t) x_{gc}(t) \tag{8}$$

where $\phi_{sgc}(t)$ represents the linear effect of combination $c \in V_g$ on source s at time t.

There are management and physical constraints on sources. If U, H and Q represent deviations from minimum maximum and cumulative limits on source inputs, then

$$U_s(t) = \mathrm{Max}(Fmin_s(t) - F_s(t), 0) \tag{9}$$

$$H_s(t) = \mathrm{Max}(F_s(t)Fmax_s(t), 0) \tag{10}$$

$$Q_s = \mathrm{Max}\left(\sum_t F_s(t) - Fcumax_s, 0\right) \tag{11}$$

The formulation could be simplified, for instance if $L_r(t)$ represented only the available water (ie the excess over $Lmin_r$) constraint (4) could be modified and a set of variables $\{M_r\}$ and Equation (5) eliminated. Unfortunately, this would result in hard constraints that are too inflexible. There may be occasions when the nominal constraints cannot be met, but schedules must always be supplied. The above formulation provides soft or elastic constraints, which can be broken at a cost. Penalties normally drive lost demand, spillage, low reservoirs and other soft infeasibilities out of the solution. Soft constraints are also vital, because it is difficult to envisage an effective metaheuristic for pump scheduling that maintains strict feasibility while exploring possible schedules.

A simple descent method

The neighbourhood of a schedule is defined to be the set of schedules that can be obtained by choosing any time-slice and any group, and changing the combination chosen at that time for that group to any other valid combination.

A simple descent method can improve schedules by systematically searching the neighbourhood of a schedule, and replacing one combination at a time by an improving combination, then searching the neighbourhood of the new schedule and so on. Descent normally finds a local optimum, which it cannot then escape. This is a drawback, but simple descent models may still be useful for making small changes to schedules, for instance when adjusting yesterday's schedule to take partial account of today's slightly different demands, or when correcting for the results of small calibration errors.

Another use of simple descent is in scheduling pumps in finer increments than a single time-slice. If pumps may be switched on or off in smaller increments, then the optimization will have more freedom, and costs may be reduced. Unfortunately, the solution time for many optimization methods is proportional to the square of the number of time-slices. A compromise is to find a near-optimal schedule with 24 time-slices, split the 24 time-slice solution into 96 × 15 min intervals, and then apply the simple descent method to obtain some advantage from the shorter on–off periods at a low computational cost. This procedure was applied to the SA results, with outcomes that will be found in Table 1.

SA and two-stage SA

Simulated annealing

In contrast to simple descent, SA has a random element that allows some non-improving changes. This enables SA to climb out of a local optimum, and eventually find a global optimum. A central part of the procedure is

Generate a new 'neighbouring' schedule at random.

Accept or reject the change at random according to the Metropolis criterion

$$\Pr(\text{accept}) = \mathrm{Min}\left[\exp\left(\frac{-\Delta\text{cost}}{T}\right), 1\right] \tag{12}$$

At high 'temperatures' (T), SA behaves like a random search, and at low temperatures, it is more like simple descent. At any given T, a series of such random changes forms a Markov chain. After sufficient changes, the distribution of costs at a given temperature will reach a thermal equilibrium, and the mean of this distribution will be an increasing function of temperature. At a given temperature, approximate equilibrium is normally reached within about Ns steps, where Ns is the neighbourhood size. When Nt is the number of time-slices and NC_g is the number of combinations for group g the size of each neighbourhood is

$$Ns = Nt \sum_g Nc_g \tag{13}$$

By starting at a very high temperature, and gradually reducing the temperature while maintaining thermal equilibrium, a global optimum will eventually be reached, as long as the neighbourhood structure allows every possible state to be reached from every other state. This process is illustrated in Figure 1, which was derived by repeating the process of SA many times. Unfortunately, keeping close enough to thermal equilibrium to guarantee optimality would take infinite time, and solution by complete enumeration would be quicker. In practice, good results can be achieved by starting at a temperature where most changes are accepted (eg point X in Figure 1), and reducing the temperature in steps of a few percent while maintaining each temperature just long enough to reach an approximate, quasi-equilibrium. In the work presented here, the length of each Markov chain at a given temperature is normally set at $2Ns$ and temperatures are reduced in steps of 5%. Annealing is terminated when no changes have been accepted for a predetermined number of trials, typically $4Ns$ (two chains).

Table 1 Comparisons of SA and LP schedules

	Time (s)	*£ Cost (24 time-slices)*	*% of LB*		
Network A: one Source, one Reservoir, four Pumps in one Group		294.96			
Lower bound					
Progressive MIP	3	295.83	100.3		
SA min		295.76	100.3		
SA mean	7.6	296.75	100.6		
SA max		297.81	100.96		
	24 time-slices			*96 time-slices*	
	Time (s)	*£ Cost*	*% of LB*	*% of LB*	*Time (s)*
Network B: two Sources, two Reservoirs, seven Pumps in three Groups					
Lower bound		1831.61			
Progressive MIP	7.3	1860.3	101.6	101	9.1
SA min		1859.13	101.5	100.7	
SA mean	13.4	1865.24	101.8	101.3	15.4
SA max		1871.77	102.2	101.8	
Network C: one Source, five Reservoirs, nine Pumps in five Groups					
Lower bound		1079.51			
Progressive MIP	10	1089.75	100.95	100.01	13
SA min		1090.25	100.99	100.1	
SA mean	29	1095.64	101.5	100.6	32
SA max		1104.12	102.3	101.3	
Network D: 13 Sources, 10 Reservoirs, 35 Pumps in 20 Groups					
Lower bound		3920.57			
Progressive MIP	233	3947.20	100.7	100.1	289
SA min		3957.21	100.9	100.3	
SA mean	588	3968.66	101.2	100.5	644
SA max		3990.34	101.8	101	

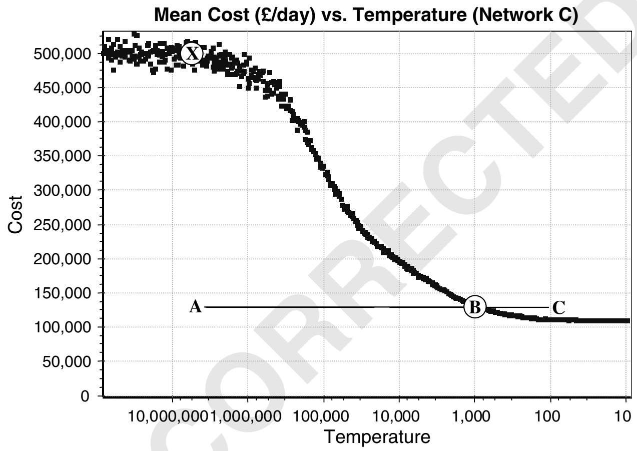

Figure 1 Annealing of a scheduling problem.

Two stage SA

Classic SA starts with a random solution or schedule. Intuitively, one might want to start with a good initial schedule. The previous day's schedule is often a good starting point. After applying the simple descent method to adjust for changes in conditions, the cost may be only 10 or 20% above the global optimum. If SA is then applied, what initial temperature should be chosen?

Points A to C in Figure 1 each have a cost that would be typical of an initial schedule obtained in this way, but different initial temperatures. At point A, the initial temperature is the same as at X, and the initial schedule will rapidly be randomized. The time taken to converge will then be the same as with the classic method, and there will be no benefit from starting with a good initial schedule. The computational time needed could be reduced by starting with a lower temperature. If the initial temperature is too low (point C), the process will be 'quenched' and the schedule will converge prematurely to a local optimum, as with descent. Suppose that a temperature can be found at which the equilibrium mean cost is very similar to the cost of the starting schedule (eg point B). Assume also that any schedule with a certain cost would in some sense be close to the centre of the equilibrium distribution with that cost, then it follows that because Markov processes have no memory, SA can start at that point and then continue while maintaining quasi-equilibrium. This is two-stage SA. Using this approach we can save a great deal of time.

Implementation

The method implemented for scheduling is based on two-stage SA. The most reliable way of setting the starting temperature is to deduce it from the typical shape of Figure 1 for the problem under consideration, which is not difficult for a routine scheduling problem. As suggested earlier, after annealing a simple descent method can be applied as a final stage to compensate for small nonlinearities and to get some benefit from reducing time-slice durations. Figure 2 summarizes the entire procedure.

Comparison of SA and LP-based results

Some results from the two-stage SA method are given in Table 1. Since there is a random element to SA, the minimum, maximum and mean costs of 10 runs of the SA method are given. Before considering nonlinear costs and constraints, it is useful to compare this local search technique with the optima that can be achieved by LP. By replacing Eqs. (5)–(7) and (9)–(11) with inequalities, and allowing $X_{gc}(t)$ to be a continuous variable representing the proportion of a pump combination to be used in a time-slice, the model described above becomes an LP. This is a linear relaxation of the discrete problem, so solution of the LP gives a lower bound to the true schedule cost. Discrete and therefore implementable solutions might approach this bound if the time-slice duration is small, but this is not guaranteed. The results of a 'progressive mixed integer' formulation are also given for comparison. This is based on LP but does not guarantee optimality.[15]

Comparisons are made for three hydraulic networks. Networks A–C are small-to-medium-sized networks. Network D, with 13 sources, 10 reservoirs and 35 fixed speed pumps is at the higher end of network sizes, though larger ones exist. The 96 time-slice results were obtained by splitting the 24 time-slice solution into 15 min intervals, then using a descent method. Direct solution with 15 min time-slices could be 16 times slower. The computer was a 1 GHz PC.

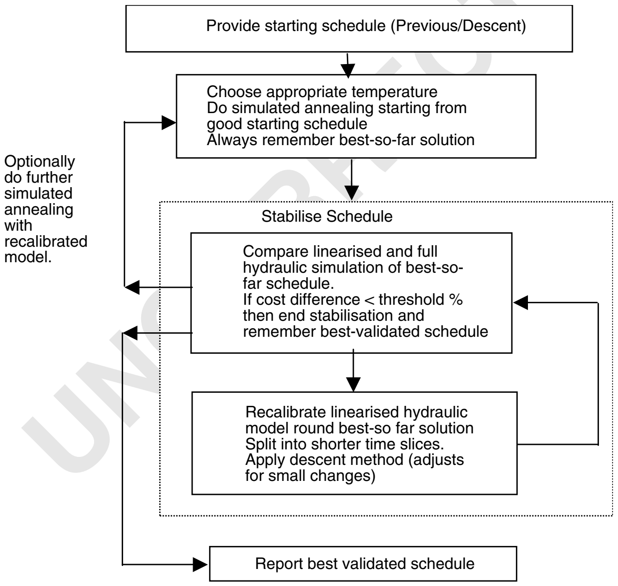
Provide starting schedule (Previous/Descent)
Choose appropriate temperature
Do simulated annealing starting from good starting schedule
Always remember best-so-far solution
Optionally do further simulated annealing with recalibrated model.
Stabilise Schedule
Compare linearised and full hydraulic simulation of best-so-far schedule.
If cost difference < threshold % then end stabilisation and remember best-validated schedule
Recalibrate linearised hydraulic model round best-so far solution
Split into shorter time slices.
Apply descent method (adjusts for small changes)
Report best validated schedule

Figure 2 A complete scheduling procedure.

The closeness of the costs in Table 1 confirms that the suggested neighbourhood structure, cooling process and two-stage methodology works well, and suggests that this approach may also give good results in cases where constraints and costs are not linear, and LP-based solutions are not available for comparison.

It can be seen that the final SA schedules are within 1.8% of the lower bound, and on average they are within 0.6% of the progressive mixed integer schedules. SA takes over twice as long as the other method, albeit using unoptimized, object-oriented code. Nonetheless, the solution time for the larger problem is suitable for practical use, and could easily be reduced by using a faster PC.

It should be borne in mind also that hydraulic models are never perfect. As much of the infrastructure is old, buried and hard to inspect, there is structural uncertainty. It is usually impossible to measure the resistance of individual pipes, and pump characteristics deteriorate in time so there is parametric uncertainty. Finally, demands cannot be perfectly predicted. These uncertainties probably give rise to errors greater than the difference between the LP and SA results.

Nonlinear cost functions

As stated above, the key advantage over MIP or LP is the ability to deal with nonlinear constraints and cost functions. Two nonlinear costs were examined—pump switching costs and maximum demand charges (MDCs).

Pump switching costs

There is a certain amount of wear and tear plus energy loss and sometimes even manual labour involved when a large water pump is switched on. Switching constraints or costs would make mixed integer formulations even less practical, and cannot be formulated in pure LP. They can be included heuristically when deriving discrete schedules from continuous results, but optimality is lost.

By contrast, it is easy to include a switching cost or penalty cost in SA and in descent methods. Figure 3 demonstrates the effect of including switching costs in the procedure described in Figure 2. The left-hand side shows a schedule obtained by SA without including switching costs. Total cost was £1104.60. Bold horizontal bars indicate periods when particular pumps are on: there are 29 distinct periods in this schedule. The right-hand side shows a schedule derived after including switching costs. Total cost is £1105.97, including switching costs of £11.70, about 1% of total cost. Other costs were little changed—in fact, in this case, they reduced a little, due to the variability of SA results. The number of distinct pump 'on' periods is just 12, which would be more acceptable in practice.

Maximum demand charges

Electricity supply utilities sometimes make significant charges for peak power consumption (measured as kVA), in order to represent infrastructure costs. These charges are applied to discrete zones, which may consist of one or more pumping stations. MDCs were added to the SA cost function, but no modification was made to the method. Figure 4 shows power consumptions for two different MDC groups. On the left, power consumption is shown for a near-optimal schedule obtained by SA with MDCs at zero. On the right, significant MDCs are included. For both MDC groups, the smoothing of power consumption is considerable. MDCs cannot be included in a pure LP because they may be incurred even if pumps are only used for a fraction of an hour. Using MIP a lower bound of £442.71 per day was determined for this example. The progressive method result was £444.51. Using SA, total cost with MDC was £455.43. The degree of difficulty in solving with MDC charges in our MIP formulation depends on whether or not the MDC zones match the pump groups and on the number of MDC zones. In some particularly difficult cases, it proved impossible even to find feasible MIP solutions for network D in less than an hour, but there was no such difficulty with SA.

Both with pump switch costs and with MDCs, it was easy to add a new cost function to the SA scheduler. The only other changes made were to the starting temperatures. This ease of formulation is one of the attractions of SA, but it should not be assumed that solving a model with nonlinear costs or constraints will always be straightforward. This is illustrated by Figure 5, which shows mean costs (£ per day) *versus* temperatures for several runs of SA for network A

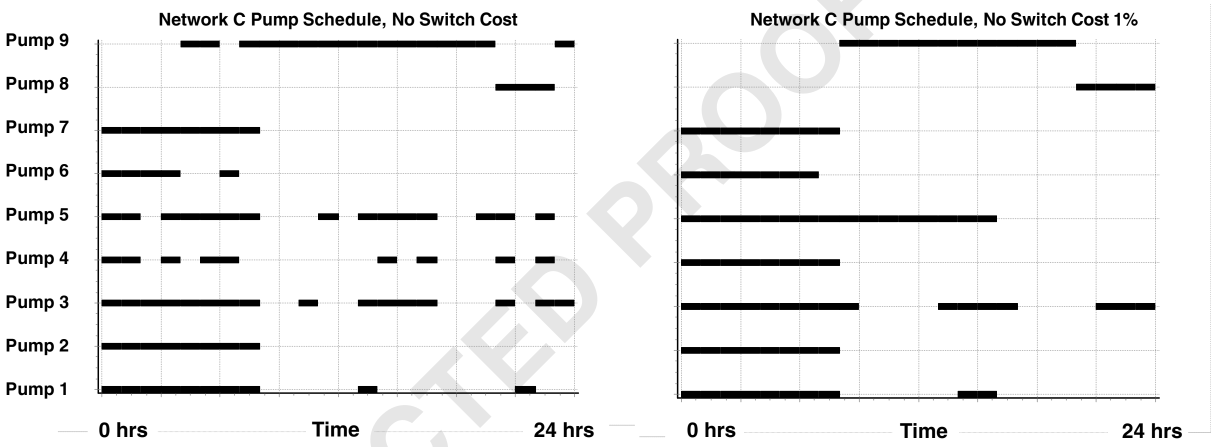

Figure 3 Schedules derived without switch costs (left) and with switch costs (right).

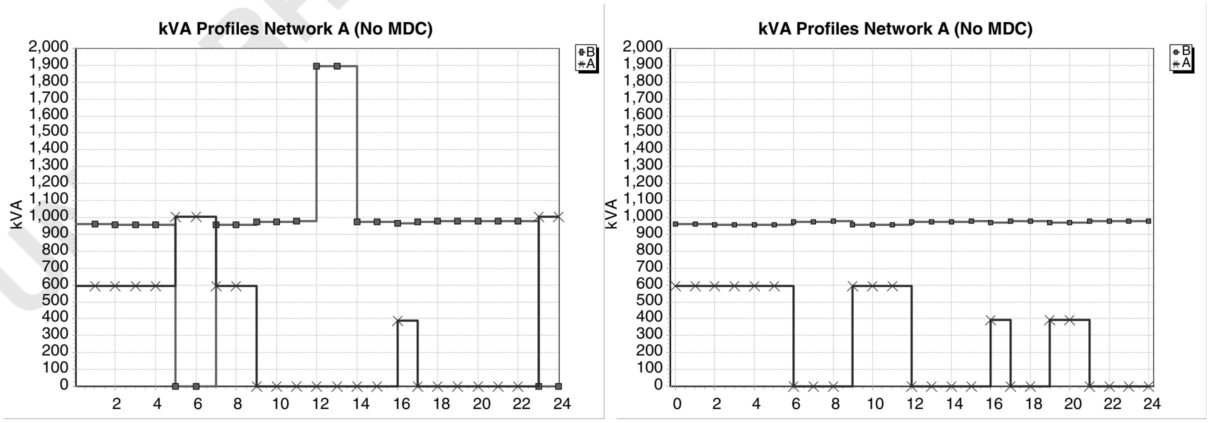

Figure 4 Power usage, without (left) and with (right) maximum demand charges (two MDC groups).

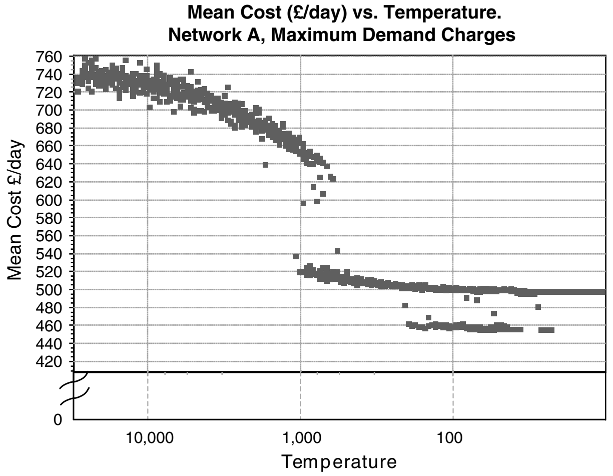

Figure 5 Mean cost *versus* temperature for repeated runs of SA, network A with MDCs.

with MDCs. The shape is unlike Figure 1. There appear to be a number of phase transitions associated with changes from one maximum kVA to another. Observation of repeated runs showed that sometimes the annealing process would be stuck in a high-cost phase for a long time before suddenly finding a much lower cost, and sometimes the lowest-cost phase might not be found at all. Reflection shows why. Assume that the schedule being perturbed has say five time-slices with a peak power requirement of 100 kVA, and that the MDC is substantial. Assume that the optimum solution would have a peak requirement of 50 kVA but higher energy use. If the kVA of any of the 5 time-slices is reduced to 50 kVA, there will be no change in the MDC. The benefit will be masked until five successive improbable changes have been made. It is only possible to reach certain states with great difficulty, and to be certain to reach the lowest-cost phase using normal SA a much slower rate of cooling is needed at certain temperatures. Paradoxically, the problem appears to be less severe in cases that are more complex, because they have many more pump combinations to choose between and therefore many more, more closely spaced kVA levels, with higher transition probabilities. These are exactly the cases that may be difficult for classical methods. Nonetheless, relative to other methods, SA results were poor for the MDC problem, even with slow cooling. A different neighbourhood structure that settles the peak power use at high temperatures and schedule timings at lower temperatures might help, but such a one has not yet been found.

Summary and conclusion

Pump scheduling is a nonlinear, non-convex partially discrete problem with large numbers of variables. Existing methods cannot guarantee optimality for non-trivial cases due to non-convexity and the need for heuristic discretization of linear solutions. This paper has shown that two-stage SA can produce near-optimal discrete schedules in a time short enough for routine operational use. Model building is based on automatic interaction with a hydraulic simulator and offers potentially wide generality and applicability. The model can be extended to deal with nonlinear effects from reservoir-level variations. The method readily allows inclusion of arbitrary nonlinear costs and constraints, which enhances the realism and acceptability of the schedules. Pump switching costs have been successfully implemented. Little effort was required to include MDCs. Poor results in this case showed that while SA will in principle work with arbitrary cost functions, it cannot be assumed that good results will always be obtained without rethinking the model or the neighbourhood structure.

Other costs and constraints warrant investigation. For example, reservoir security is a complex nonlinear function of the diurnal reservoir profile. There should be benefit in using a direct constraint on security rather than the conventional proxy, which is a simple lower limit on level.

There is scope for further work on the automatic setting of start temperatures for two-stage SA. There is no problem in routine scheduling, but an efficient method for new or out of the ordinary circumstances would be useful.

A wide variety of sophisticated cooling schedules has been discussed in the literature, for example Li *et al.*[16] One of them might provide further time savings or improved optimality. Alternatively or additionally penalty charges could be varied with temperature. These possibilities might be particularly relevant when MDCs are included.

Notation

C	pump combination index
g	group index
r	service reservoir index
s	source index
t	time-slice index
x	proportion of a combination used in the schedule
A	unmet demand
B	spillage
C	total cost
D	time-slice duration
F	flow from source
Fcumax	maximum permitted daily source output
Fmax	maximum permitted flow rate
Fmin	minimum permitted flow rate
H	deviation above maximum permitted source flowrate
K	deviation from end-target reservoir level
L	service reservoir level
Lfull	physical maximum reservoir level

Lmax	permitted maximum reservoir level
Lmin	minimum permitted reservoir level
Ltarget	desired end-of-day reservoir level
M	deviation below minimum permitted reservoir level
Nt	number of time-slices
Ns	neighbourhood size
Nc	number of combinations
O	deviation above maximum permitted reservoir level
P	penalty cost
Q	deviation above maximum permitted daily source output.
T	annealing temperature
U	deviation below minimum permitted source flow rate
V_g	set of admissible combinations of switched on pumps in group *g*
ε	cost impact (energy cost per unit time)
ϕ	source impact (contribution to flowrate)
ρ	reservoir impact (contribution to rate of change of level)

Acknowledgments—We thank the anonymous referees for their valuable comments and suggestions, which have greatly improved this paper.

References

1 OFWAT (2002). June Return Table 21.

2 Sterling MJH and Coulbeck B (1975). A dynamic programming solution to optimization of pumping costs. *Proc Inst Civ Engrs* **59**(2): 813–818.

3 Andersen JH and Powell RS (1999). The use of continuous decision variables in an optimising fixed speed pump scheduling algorithm. In: Powell RS and Hindi KS (eds). *Computing and Control for the Water Industry*. Research Studies Press, pp 119–128. Q1

4 Ulanicki B, Coulbeck B and Ulanicka K (1999). Generalised techniques for optimisation of water networks. In: Powell RS and Hindi KS (eds). *Computing and Control for the Water Industry*. Research Studies Press, pp 163–175. Q2

5 Jowitt PW and Germanopoulos G (1992). Optimal pump scheduling in water supply networks. *J Water Resour Plan Mngt ASCE* **118**(4): 406–422.

6 Burnell D, Race J and Evans P (1993). The trunk scheduling system for the London water ring main. In: Coulbeck B (ed). *Integrated Computer Applications in Water Supply Volume 2: Applications and Implementations for Systems Operation and Management*. Research Studies Press, pp 203–217. Q3

7 Ormsbee L and Lansey K (1994). Optimal control of water supply pumping systems. *J Water Resour Plan Mngt, ASCE* **120**(2): 237–252.

8 Walters GA, Savic DA, Thurley RWF, Halhal D, Kapelan Z and Atkinson R (1999). Optimal design of water systems using genetic algorithms. Some recent developments. In : Powell RS & Hindi KS (eds). *Computing and Control for the Water Industry*. Research Studies Press, pp 337–344. Q4

9 Sousa J and da Conceicao Cunha M (1999). On the quality of a SA algorithm for water network optimisation problems. In: Savic DA and Walters GA (eds). *Water Industry Systems: Modelling & Optimisation Applications, Volume 2*. Research Studies Press, pp 333–345. Q5

10 Mackle G, Savic DA and Walters GA (1995). Application of genetic algorithms to pump scheduling for water supply. In: Genetic Algorithms in Engineering Systems: Innovations and Applications, GALESIA '95, IEE Conference Publication No. 414, Sheffield, UK, pp 400–405.

11 Goldman F and Mays L (1999). The application of simulated annealing to the optimal operation of water systems. In: Wilson EM (ed). Proceedings of the ASCE 26th Annual Water Resources Planning and Management Conference, CD-ROM.

12 Varanelli JM (1996). *On the acceleration of simulated annealing, Chapter 3. A two stage simulated annealing methodology*. PhD dissertation, University of Virginia.

13 Rossman LA (1994). Epanet Users Guide, Drinking Water Research Division, Risk Reduction Engineering Laboratory, Office of Research and Development, US Environmental Protection Agency, Cincinnati, Ohio.

14 Ulanicki B and Orr CH (1991). Unified approach for the optimization of nonlinear hydraulic systems. *J Optim Theory Appl* **68**(1): 161–179.

15 McCormick G and Powell RS (2003). A progressive mixed integer programming method for pump scheduling. In: Maksimovic C, Butler D and Memon FA (eds). Advances in Water Supply Management (Proceedings of CCWI 2003), AA Balkema Publishers, Rotterdam, pp 307–313.

16 Li YH, Richards EB, Liang YJ and Azarmi NAS (1996). Localized simulated annealing. In: Rayward-Smith VJ *et al* (eds). *Constraint Satisfaction and Optimization. In Modern Heuristic Search Methods*. Unicom Wiley, pp 27–39. Q6

Received July 2003;
accepted December 2003 after one revision