On solving water distribution network design problems with stochastic search optimization techniques

Joaquim Sousa & Maria da Conceição Cunha
Superior Institute of Engineering, Polytechnic Institute of Coimbra, Quinta da Nora - 3030 Coimbra - Portugal
EMail: jjoseng@sun.isec.pt, mccunha@sun.isec.pt

Abstract

In spite of twenty years of research efforts, optimal design of looped water distribution systems still remains a challenge. The physical behaviour of a looped network is described by a set of equations, some of them non-linear. Although numerical methods can simulate the hydraulic behaviour of large size looped networks, finding the optimal network design is, even for simple networks, a complex task. If we are to represent reality adequately, this problem must be formulated in combinatorial form defining extremely difficult models, whose resolution in real-world conditions, via classical optimization techniques, is often impossible. However new developments in the field of stochastic optimization techniques allow their solution to be faced with some hope of success.

In this paper a simulated annealing algorithm is proposed to solve this kind of problem. Simulated annealing is a stochastic search optimization method that can work well for large-scale optimization problems that are cast in discrete or combinatorial form. The authors present some work on the application of this heuristic method to three well-known case studies (Alperovits and Shamir, Hanoi, and New York networks). The results obtained show an improvement both on the optimal solution and on computer running time when compared with those from the literature. This fact was encouraging, leading us to deal with more complex systems in this paper. The results found through this study can be considered very promising.
1 Introduction

The complexity of the problem of determining the least-cost design of a looped water distribution network is very high. The corresponding mathematical models are NP-hard, thus very difficult to handle with traditional optimization methods. For comprehensive reviews of the application of traditional optimization methods (linear, non-linear and dynamic programming) see Lansey and Mays,^9^ and Simpson et al..^13^ More recently, developments in the field of stochastic search optimization have allowed the resolution of design optimization problems formulated as non-linear mixed-integer problems. A great amount of research has been conducted in this field, and new techniques, like genetic algorithms and simulated annealing, could play an important role in dealing with the discrete nature of the corresponding problems. For a presentation of genetic algorithms see Walters and Cembrowicz,^14^ Simpson et al.,^13^ Dandy et al.,^4^ and Savic and Walters.^12^

In this paper, a simulated annealing heuristic is proposed to solve the aforementioned problem. Simulated annealing heuristics have proved to be extremely efficient techniques in solving classic combinatorial problems such as the travelling salesmen problem. The popularity of simulated annealing is growing and new areas of application are being reported every day. In order to evaluate the performance of this heuristic method, well-known networks taken from the literature were solved (Alperovits and Shamir,^2^ Hanoi, and New York networks). The results obtained, as compared to those obtained by other optimization methods, confirm the ability of the heuristic to handle this type of non-linear mixed-integer problem. The heuristic has performed extremely well for the three standard test problems, providing a discrete diameter solution, more realistic than continuous or split-pipe designs proposed as optimal solutions by other authors. The good performance of the algorithm proposed encourages us to engage with more realistic but more complex problems, like those including pumps and reservoirs operating under multiple loads. After describing an overview of the heuristic and its features, and showing a summary of the solutions obtained for Alperovits and Shamir,^2^ Hanoi and New York networks, we present the results obtained for a more realistic case study.

2 The simulated annealing algorithm

Simulated annealing is a heuristic method that can work well for large-scale optimization problems that are cast in discrete or combinatorial
form. This heuristic seeks to overcome major shortcomings of classical optimization methods in searching for global optimality. Traditional optimization techniques accept only improving moves, while annealing algorithms search the solution space more slowly, allowing worsening moves to be accepted in order to escape local optima. The design of this heuristic is based on an analogy with the physical process representing the way solids cool and anneal. In this physical process, first the temperature is increased to allow mobility of the molecules. Later, the temperature is carefully decreased and then the molecules will arrange themselves randomly until the lower energy state is reached, which corresponds to the crystalline structure. The rapid decreasing of temperature would not provide final crystalline structure because low energy states would be reached too quickly. Metropolis et al., introduced a simple algorithm that expresses the ideas that have been described. The sequence of states (or configurations) generated by the Metropolis algorithm is based on a Monte Carlo technique. Supposing that the energy of the current state is \( E_i \), a perturbation mechanism is applied to generate a state \( j \), which energy is \( E_j \). If \( E_i - E_j > 0 \), \( j \) will be the new current state. Otherwise (according to Metropolis criterion) \( j \) will be accepted as the new current state with a probability given by \( p = \exp\left\{ (E_i - E_j) / k_B t \right\} \) \((k_B: \text{Boltzmann constant}; t: \text{temperature})\). The temperature is used to control the probability of accepting worsening moves, and it will be lowered as the algorithm develops. For large values of temperature, most uphill moves will be accepted; as the temperature level approaches zero more and more uphill moves will be rejected. If the temperature decreases very quickly, the situation observed with traditional optimization techniques will happen: premature convergence to local optima. The mobility in this search improves the likelihood that all possible regions of the solution space can be reached.

The analysis of the annealing algorithm in terms of Markov chains, providing that certain qualifications are satisfied, permits finding general convergence results (Aarts and Van Laarhoven).

In general terms, an annealing algorithm will include the following steps:

1. choose \( s_i \) \{\( s_i \) is the initial configuration\}
2. choose \( t_i \) \{\( t_i \) is the initial temperature\}
3. choose \( t_f \) \{\( t_f \) is the stopping temperature\}
4. \( j \leftarrow 0 \)
5. repeat
6. \( j \leftarrow j + 1 \)
7. choose at random \( s'_j \in N(s_j) \) \{\( N(s_j) \) is the candidate set of \( s_j \}\}
8. choose at random \( p \in [0, 1] \)
9. if \( p \leq \min\{1, \exp\left(\frac{c(s_j) - c(s'_j)}{t_j}\right)\} \)
   \{\( c(s_j) \) is the cost of the current configuration\}
   \{\( c(s'_j) \) is the cost of the candidate configuration\}
   then \( s_{j+1} \leftarrow s'_j \)
   else \( s_{j+1} \leftarrow s_j \)
10. choose \( t_{j+1} \leq t_j \)
11. end

The implementation of annealing algorithms comprises two main aspects: a perturbation mechanism and a cooling schedule.

The perturbation mechanism (see Lin,\(^1\) Kirkpatrick et al.,\(^8\) Aarts and van Laarhoven\(^1\)) defines how the current solution is randomly changed in order to obtain a candidate solution that will be evaluated using the Metropolis criterion. The perturbation mechanism employed here consists of choosing a decision variable (the diameter of a pipe, or the water level in a reservoir, or the head added by a pump) whose value will be changed at random. The value of that variable will be increased in 40% of the cases, and in the other 60% decreased.

The cooling schedule defines how to handle the parameter temperature. After proposing an initial temperature, it must provide the way how and when such temperature should be decreased along the implementation. A stopping criterion is also included. The cooling schedule comprises four parameters (Johnson et al.\(^6\)):

- \( \alpha \): the probability of accepting a transition from the initial configuration to a candidate configuration whose cost is superior to that of the initial configuration in a given percentage. This parameter, called elasticity of acceptance, is used to define the initial temperature of the annealing process. This can be done using the following expression:
  \[ t_i = -0.1 \frac{c_0}{\ln(\alpha)} \] (\( c_0 \): cost of the initial configuration; \( \alpha \): elasticity of acceptance). This expression allows finding the temperature for which \( \alpha \)% of solutions, with cost 10% higher than the cost of the initial configuration, will be accepted.
Hydraulic Engineering Software

- $n_1$: the minimum number of algorithm iterations that will be performed even without an improvement of the optimum or the average solution cost, before decreasing temperature. The simulated annealing algorithm should reach equilibrium at each temperature before progressing to the next temperature. This parameter will have a decisive influence upon the computation time and the rate of convergence.

- $r$: the rate at which temperature is decreased, whenever a temperature decrease should occur.

- $n_2$: the number of temperature decreases that will be performed without an improvement of the optimum or of the average solution cost, before stopping the algorithm.

3 Three standard test networks

The simulated annealing algorithm was initially used to solve three standard test problems of the literature: Alperovits and Shamir\(^2\) (simple example), Hanoi, and New York networks. The general statement of the optimization problem to be solved is as follows: choose from a discrete commercial diameters set the combination that gives rise to the least-cost network necessary to supply a set of demand nodes within a prescribed range of pressures. The set of constraints to add include the energy and mass conservation laws, with head losses determined by the Hazen-Williams equation. Steady state flow conditions were assumed and a node formulation was used to represent the hydraulic behaviour of the network.

In the design of any heuristic, computational effort and solution quality are important criteria. Therefore, an appropriate number of runs were performed to obtain the set of annealing parameters that works best considering those two criteria. The value found for the elasticity of acceptance was $a = 0.01$. The values of the cooling factor and minimum number of iterations to be performed at each temperature can change, as the algorithm develops, according to the percentage of solutions accepted. The initial values for these parameters were: $r = 0.8$ and $n_1 = 10$. The percentage of solutions accepted is also used to define the number of temperature decreases that will be performed without an improvement of the optimum or the average solution cost before stopping the algorithm.

As this method is essentially a random search method, in order to gain more confidence in the solutions found, an extra 100 runs, for each network, were performed with different seed numbers for the pseudorandom generator. The results obtained are summarised in Table 1.
Table 1: Solutions summary of the three standard test networks.

<table>
<thead>
<tr>
<th></th>
<th>Alperovits and Shamir (1977)</th>
<th>Hanoi</th>
<th>New York</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal solution cost</td>
<td>419,000</td>
<td>6,056,370</td>
<td>37,130,410</td>
</tr>
<tr>
<td>Optimal solutions</td>
<td>80%</td>
<td>70%</td>
<td>60%</td>
</tr>
<tr>
<td>Distant by less than 1%</td>
<td>20%</td>
<td>22%</td>
<td>36%</td>
</tr>
<tr>
<td>Distant between 1 and 4%</td>
<td>0%</td>
<td>8%</td>
<td>4%</td>
</tr>
<tr>
<td>Average CPU time/run</td>
<td>22 sec.</td>
<td>482 sec.</td>
<td>256 sec.</td>
</tr>
</tbody>
</table>

Those results were found using a value of 10.5088 for the conversion factor of the Hazen-Williams equation. As Savic and Walters,\(^2\) pointed out, there is a large range of variation of this factor in the different case studies presented in the literature. Accordingly, the three problems were solved considering the upper bound for this factor (10.9031) and the value used by Alperovits and Shamir\(^2\) (10.6792). Alperovits and Shamir\(^2\) network has the same cost for all the three values. The simulated annealing solutions cost of the Hanoi network for conversion factor values of 10.9031 and 10.6792 were 6.183 and 6.093 respectively (cost in million $). The solutions of the New York network are similar to those presented by Savic and Walters,\(^2\) and Dandy \textit{et al.}\(^4\) (for the same values of the conversion factor of the Hazen-Williams equation used by these authors).

The diameters and pressure (always above the 30 m required) of Alperovits and Shamir\(^2\) and Hanoi networks determined by simulated annealing were presented in Cunha and Sousa\(^3\).

A comparison of the results indicates that simulated annealing performs very favourably vis à vis other optimization methods reported in the literature, both in terms of solution time and quality.

With respect to the genetic algorithm solution of the literature (Savic and Walters,\(^2\) and Dandy \textit{et al.}\(^4\)), we observed improvements in both solution quality (for the Hanoi network) and computing effort.

### 4 Complex two-loop network

The addition of other aspects such as reservoirs having operational storage (balancing reservoirs), and pumps operating under multiple loading conditions to the simple networks considered before is very important in order to have an algorithm capable of handling real-world problems. This addition will, undoubtedly, increase the complexity of the
problem, but the good performance of the algorithm proposed encouraged us to engage in such a task.

To illustrate this kind of application, we solved the second example presented in Alperovits and Shamir\(^2\) (Figure 1). All data are from that example. The network comprises nine pipes, one source, one balancing reservoir and a pump. There are two load conditions, one corresponding to peak hours when the balancing reservoir contributes to supply the network (daytime) and the other corresponding to no-demand hours when the balancing reservoir is filled up (night-time).

Each configuration evaluated by the simulated annealing algorithm is defined by the diameters for each pipe, the balancing reservoir level and the head added by the pump. An initial configuration is proposed. This configuration is randomly changed using the perturbation mechanism presented before. All the constraints must be observed for the two loads considering this new configuration (head added by the pump can be different for each load). Two different scenarios regarding pressure requirements were used, one considering minimum pressure requirements of 30 m (scenario 1) and the other considering additional maximum pressure requirements of 70 m (scenario 2).

Pipe diameters and node pressures for each scenario obtained through simulated annealing are shown in Table 2. In the first scenario, the pumps operating under daytime and night-time loads added, respectively, 24.082 m and 68.481 m to the head. In the first scenario the total cost comprises the pipeline cost (168,000), the cost for raising the balancing reservoir by 7.71 m above the ground level (15,420), and the pumping cost (50,005). In the second scenario, the pumps operating under daytime and night-time loads added, respectively, 12.493 m and 20.059 m to the 210 m

\[
\text{Reservoir 700.00} \quad 203.21m
\]

\[
\text{247.89} \quad 1273.93
\]

\[
\text{Pump 300.00} \quad 203.21m
\]

\[
\text{1273.93} \quad 273.93
\]

\[
\text{Source 300.00} \quad 203.21m
\]

\[
\text{1273.93} \quad 273.93
\]

\[
\text{Reservoir 700.00} \quad 203.21m
\]

\[
\text{247.89} \quad 1273.93
\]

\[
\text{Pump 300.00} \quad 203.21m
\]

\[
\text{1273.93} \quad 273.93
\]

\[
\text{Source 300.00} \quad 203.21m
\]

\[
\text{1273.93} \quad 273.93
\]

\[
\text{Reservoir 700.00} \quad 203.21m
\]

\[
\text{247.89} \quad 1273.93
\]

\[
\text{Pump 300.00} \quad 203.21m
\]

\[
\text{1273.93} \quad 273.93
\]

\[
\text{Source 300.00} \quad 203.21m
\]

\[
\text{1273.93} \quad 273.93
\]

Figure 1: Network with a pump, a balancing reservoir, loads, and final flow distribution (scenario 1).
head. In the second scenario the total cost comprises the pipeline cost (203,000), the cost for raising the balancing reservoir by 5.64 m above the ground level (11,280), and the pumping cost (40,581). Table 3 presents results with respect to the total cost of each scenario, as well as some results reported in the literature. The solution from Kessler and Shamir, was obtained considering only the daytime load. The solution improvement resulting from the utilisation of the simulated annealing should be emphasised. The computing running time was about 100 seconds on a Pentium PC at 166 Mhz.

Table 2: Optimal solution of complex two-loop network.

<table>
<thead>
<tr>
<th>Diameters (in.)</th>
<th>Pressures (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipe</td>
<td>Scen. 1</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 3: Summary of solution costs.

<table>
<thead>
<tr>
<th>Total Cost</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alperovits and Shamir (1977)</td>
<td>299,851</td>
<td>291,079</td>
</tr>
<tr>
<td>Eiger et al. (1994)</td>
<td>254,861</td>
<td>254,861</td>
</tr>
</tbody>
</table>

5 Conclusions

The complexity of the problem of determining the least-cost design of a looped water distribution network is very high. Simulated annealing is a very promising stochastic search method that provides a means for avoiding local optima. Its application to the aforementioned problems, even for more realistic situations like those including pumps, balancing reservoirs and multiple loading conditions, presents good results in comparison to other optimization methods. Through the applications
developed in this paper, its good performance vis à vis genetic algorithms, considering both computer effort and solution quality have been shown.

In future research, a large number of problems representing a variety of situations will be solved in order to evaluate the consistency and the robustness of annealing parameters set. In fact, three problems are not enough for general conclusions to be drawn. Furthermore, our future research will deal with problems presenting all the complexity of real-world situations.

References


150 Hydraulic Engineering Software


