Comparison of deterministic and heuristic optimization solvers for water network scheduling problems
J. G. Bene, I. Selek and Cs. Hős

ABSTRACT
This paper presents a novel approximate dynamic programming technique for solving the pump schedule optimization of real-water distribution networks. The method is based on the significant decreasing of the search space by splitting the optimization problem into smaller units. In addition, the state space of the main distribution system was further reduced to the most important reservoirs. The capabilities of the proposed technique are demonstrated on a real-life problem, the water distribution system of the town of Sopron, Hungary. Nine test cases were defined which represent different initial water level scenarios, thus the new application was easy to compare to a former developed genetic algorithm and to some world-leading optimization solvers which are available on the NEOS Server. The benefits and drawbacks of these deterministic and heuristic methods are highlighted.

Key words | approximate dynamic programming, genetic algorithm, pump schedule optimization

INTRODUCTION
There has been much research into the pump schedule optimization of water distribution systems due to the high demand for saving energy or total working expenses. The storage capacity of the reservoirs provides the possibility that domestic and industrial water consumers can be satisfied with different pump schedules. This storage capacity can be exploited for different purposes which are the following: (a) if the energy tariff changes during the optimization horizon the reservoirs can be filled at the lower tariff and the consumers can be satisfied from the reservoirs during peak hours; and (b) if the energy tariff is uniform the pumps’ specific energy consumption (the energy demand for conveying a given amount of fluid, [kWh/m³]) is a good measure which describes the system efficiency well, see Wallbom-Carlson (1998) and Bene & Hős (2012). Usually bigger pumps have lower specific energy consumption thus it is more economical to use them and store the ‘unnecessary’ amount of water in the reservoirs. Although these rules of thumb seem simple, due to the complexity of the system the optimization problem is hard to solve which is mainly due to the large variety of the constraints and the mixed-integer type of the optimization problem.

Evolutionary computing methods are widely applied to solve the above described problem due to their flexibility and robustness: Bene et al. (2010), Bene & Hős (2010), Barán et al. (2005), Kadu et al. (2008) and Selek et al. (2012). The drawback of these methods is that even one single run requires plenty of evaluation of the objective function (e.g. a hundred thousand) without any guarantee of the quality of the results. In the case of industrial applications the reliability of the algorithm is essential because there is no time to run more computations: every single run must provide ‘good’ quality which satisfies at least the constraint system – in the case of evolutionary methods it is not guaranteed.

Among deterministic solvers, dynamic programming is a popular optimization solver for pump schedule optimization, however, the available literature mostly deals with continuous flow rate pumps and discrete pumps are rarely used: Chandramouli & Raman (2001), Kumar & Baliarsingh (2005) and Selek et al. (2012). Another problem is that the method suffers from the ‘curse of dimensionality’ (Powell 2007) which means that the computational demand grows exponentially with the problem size which makes the optimization of real-size networks computationally unfeasible.
The aim of this paper is to present a novel approximate dynamic programming method (abbreviated to ADP) for solving mixed-integer pump scheduling problems (both discrete and continuous flow rate pumps in the network). The approach must guarantee a success rate of 100% while the provided objective values are not worse than the ones gained by an up-to-date genetic algorithm (abbreviated to GA).

This paper is organized as follows. The following section describes the mathematical model of water distribution systems and introduces some simplification. Then follows the description of the possible methods, e.g. a traditional dynamic programming method, the developed novel method, the former developed GA and some other state-of-the-art solvers. In the next section the water distribution system of Sopron is introduced in detail. Then begins the presentation of the obtained results: a comparison of the objective values of the different solvers is given and two optimized schedules are presented in order to investigate some practical issues. The last section makes conclusions and discusses the possible future research directions.

**OPTIMIZATION MODEL OF WATERWORKS USING ‘FLOW ONLY’ MODEL**

In this section the mathematical description of the optimization problem is given. The applied hydraulic model is known as a ‘flow only’ model in the literature (Cembrano et al. 2000) since the pumps’ operating points are considered as a priori known, constant values and only the mass conservation law is used for computing the flow in the pipes. Note that a discussion about the possibility of using other hydraulic models is given in the last section.

The network to be optimized may consist of the following elements: water sources, variable speed well pumps, fixed speed pumps, water reservoirs, water demands and pipelines (for an example see Figure 1).

The time horizon of the optimization problem includes the next 24 hours. The operating points are not allowed to be switched ‘too often’, therefore the time scale is usually divided into 1 hour long optimization periods which assures the above condition:

\[ \Delta t_i = 1[h], \quad i = \{1, 2, \ldots, 24\}. \]

The pump flow rates are the control variables of the system. In this paper the term ‘pump’ or ‘constant speed pump’ refers to a discrete pump or a pump group which consist of more parallel switched discrete pumps. Thus the pumps have two or more discrete states, i.e. well defined flow rate – power consumption pairs. The flow rate and the power consumption of the \( j \)th pump in the \( i \)th time period is the following:

\[ \{ q_{pump}^{i,j}, P_{pump}^{i,j} \}. \]

The pumps are connected to power supplies whose power output is not allowed to exceed a given limit:

\[ \sum_{k \in P_j} P_{pump}^{i,k} \leq P_{pump}^{max}, \]

where \( P_j \) is the set of the connected pumps to the \( j \)th power station. The so-called ‘well pumps’ (or ‘variable speed pumps’) play a unique role in the system. The flow rate of the \( j \)th well pump can be set continuously within a given range

\[ q_{well}^{min,j} \leq q_{well}^{i,j} \leq q_{well}^{max,j}, \]

but their operating points can be changed only at a few (given, typically two to five) time instances per day due to technological reasons:

\[ q_{well}^{i,j} = q_{well}^{i+1,j} \quad \text{if} \quad i \in F, \]

where \( F \) is the set which describes the periods when the change is not allowed. There are also daily minimum and maximum limits for the exploited water for each well:

\[ V_{well}^{min} \leq \sum_{i=0}^{23} q_{well}^{i,j} \Delta t_i \leq V_{well}^{max}. \]
The power consumptions of the well pumps are negligible compared to the other pumps. The continuity laws determine the water levels in the $j$th reservoir:

$$V_{i+1,j} = V_{i,j} + \left( \sum_{k \in P^+} q_{i,k}^{\text{pump}} - \sum_{k \in P^-} q_{i,k}^{\text{pump}} + \sum_{k \in W_i} q_{i,k}^{\text{well}} \right) \Delta t_i - \sum_{k \in D_i} d_{i,k},$$

where $i = \{0,1, 2, \ldots, 24\}$ and $d$ represents the water demands and inflows (W3 and D4-D8 in Figure 1) which are considered deterministic and known such as the initial water volumes $V_{0,j}$. The reservoir volumes must be in a given, time dependent range:

$$V_{ij}^{\text{min}} \leq V_{ij} \leq V_{ij}^{\text{max}}.$$

Figure 1 | The model of the water distribution system of Sopron.

The power consumptions of the well pumps are negligible compared to the other pumps. The continuity laws determine the water levels in the $j$th reservoir:

$$V_{i+1,j} = V_{i,j} + \left( \sum_{k \in P^+} q_{i,k}^{\text{pump}} - \sum_{k \in P^-} q_{i,k}^{\text{pump}} + \sum_{k \in W_i} q_{i,k}^{\text{well}} \right) \Delta t_i - \sum_{k \in D_i} d_{i,k},$$

where $i = \{0,1, 2, \ldots, 24\}$ and $d$ represents the water demands and inflows (W3 and D4-D8 in Figure 1) which are considered deterministic and known such as the initial water volumes $V_{0,j}$. The reservoir volumes must be in a given, time dependent range:

$$V_{ij}^{\text{min}} \leq V_{ij} \leq V_{ij}^{\text{max}}.$$
Time dependence of the reservoir limits is important because they are typically narrower in the last period of the optimization horizon (they are set within a given range from the initial water volumes).

The above mentioned equations describe the dynamics and the constraints of the system while the objective function is the total cost:

$$\text{objective}_1 = \sum_{i=0}^{23} \sum_{j \in \text{pumps}} \Delta t_j c_i,$$

where $c_i$ is the energy charge [€/kWh] in period $i$. Due to the industrial needs the number of pump switches could serve as an alternative objective function:

$$\text{objective}_2 = \sum_{i=0}^{22} \sum_{j \in \text{pumps}} \text{sign}(q_{ij} - q_{i+1,j})^2.$$

### SOLUTION METHODS

This section deals with a few possible solution techniques for the above described optimization problem. Although the focus is on the novel ADP technique, other solvers are also presented which are used for comparison.

#### Dynamic programming approaches

Dynamic programming approaches are widely used in order to reduce the problem size. These methods work on the state space $x_i$ and the action space $u_i$. In this application the state space involves the water levels and the pumps' flow rates correspond to the action space while $i$ is the index for the time steps:

$$x_i = [V_{i,1}, V_{i,2}, \ldots, V_{i,\text{res}}]$$

$$u_i = [q_{i,1}^{\text{pump}}, q_{i,2}^{\text{pump}}, \ldots, q_{i,\text{pump}}, q_{i,1}^{\text{well}}, q_{i,2}^{\text{well}}, \ldots, q_{i,\text{well}}].$$

The dynamic of the system (which was described by (1)–(3), (7)) can be compressed to the function $f$:

$$x_{i+1} = f(x_i, u_i, d_i).$$

If the value (optimal cost) of being in state $x_i$ is $J_i(x_i)$ and the cost of the action $u_i$ is $c^a_i(x_i, u_i)$ the optimization problem can be solved recursively using the Bellman equation (Powell 2007):

$$J_i(x_{i+1}) = \min_{u_i} [J_i(x_i) + c^a_i(x_i, u_i)].$$

### Splitting the model into subproblems

One of the main ideas of the novel method is splitting the network into smaller units in order to reduce the state and the action space of the solvable sub-models compared to the original one. The remaining sub-models are the main distribution system and the well fields (an example is shown in Figure 1); each well field consists of the well itself, the first reservoir after the well and the variable speed pump which delivers the water between them. The developed algorithm is the following.

The core of the algorithm is a standard dynamic programming approach; the only difference is that it works only on the main distribution system. If one of the solutions does not satisfy constraints (3) or (8) it will be removed just like in the original algorithm. Then, the fulfilling of the well
constraints (4)–(6) and the reservoir constraints of the well field are checked separately as follows. For each well field a linear programming (LP) problem can be defined which incorporates these constraints. The outgoing pump schedule \( q_{\text{pump}, \text{outgoing}} \) (the forward delivered water from the well field) is known in a given cell (i.e. candidate solution) of the computational grid, thus the following equation system can be written in the \( i \)th period for the \( j \)th well field:

\[
V_{\text{min}, i + 1,j} \leq V_{0,j} - \sum_{t=0}^{i} q_{\text{pump}, \text{outgoing}} + \sum_{t=0}^{i} q_{\text{well}} \leq V_{\text{max}, i + 1,j} \text{ for all } t \in \{0, 1, \ldots, i\}
\]  

(15)

The above equation system is a specific form of Equation (8) and its variables are only the well flow rates. Constraint (4) can be easily satisfied by setting the limits for variable \( q_{\text{well}} \). The daily amount constraint could be inserted directly into the problem in the same form as in (6). Constraint (5) allows switching the flow rates of the wells only a few times a day. Let these time instances be 7, 1, 5 and 8 a.m. Then using the constraint new variables must be introduced as follows:

\[
q_{\text{well}} = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_11, v_12, v_13, v_14, v_15\},
\]

(16)

where \( v_0 \) is a fixed, initial flow rate. The advantages of the above described method are obvious: both the state space and the action space of the remaining subsystems were decreased. Instead of discretizing the variable speed well pumps, a few LP problems are solved. The only restriction is that the power consumption of the well pumps must be negligible compared to the other ones (or it must be approximated by a linear function of the flow rate).

Selecting the ‘key reservoirs’

Although the above described method decreased the problem size drastically, it is still too high for on-line computations, thus a further approximation was introduced. The idea is to diminish the state space to only a few (e.g. two) reservoirs, which are called key reservoirs and indexed with \( k_0 \) and \( k_1 \):

\[
x_i = [V_{i,k_0} \ V_{i,k_1}]
\]

(17)

The action space remains the same which allows for computing the actual volume of all reservoirs in order to check constraints; but only two reservoirs are used for making a computational grid for the dynamic programming. The benefit is easy to understand: the state space is reduced to \( g^2 \) where \( g \) is the number of cells within one reservoir. However, selecting the key reservoirs is a challenging task and highly network dependent. An example method will be shown in the application section.

The pseudo code of the novel ADP algorithm (which exploits both above described techniques) is given as follows:

```
set the initial state
for \( t = 1 \) to \( T \) do
    for all state built from the key reservoirs (KR) do
        for all possible control action of the main distribution system (MDS) do
            compute the new state of the KR
            if the MDS is feasible then
                solve all LP-subproblems which describe the well fields (WF)
                if all WF is feasible then
                    check the cost and write the new solution if necessary
                    ...
                end if
            end if
        end for
    end for
end for
choose the best solution
```

Other solvers

Although the above described ADP method is able to solve the optimization problem, the question arises as to how far the obtained results are from the global optimum. Up to now, no technique exists which is capable of finding the
global optimum thus the best option is to solve the test problems with other (both deterministic and stochastic) solvers and compare the results. For a stochastic solver, a formerly developed up-to-date GA (Selek et al. 2012; Bene et al. 2010; Bene & Hős 2010) was used. In addition, six state-of-the-art general purpose deterministic solvers were applied. These solvers are available on the internet (NEOS 2013) and can solve optimization problems which are described in the AMPL modelling language.

APPLICATION: WATERWORKS OF SOPRON

The application example of this work is the waterworks of Sopron city, located in west Hungary and depicted in Figure 1. The waterworks supply about 60,000 domestic consumers and some significant industrial demands, e.g. a brewery. Due to the hilly terrain of the city the water level variation in the reservoirs and the friction losses are negligible compared to the geodetic heights thus the flow rate–power consumption function of the pumps can be determined a priori. It means that there is no need for coupled hydraulic simulation during the optimization process; therefore the former introduced flow only model can be used for the computations. (Note that this assumption can be used in the case of the majority of the regional waterwork systems.)

The system data can be downloaded from Bene (2013). Nine different test cases were defined which represent different initial water volume scenarios.

Applying the novel ADP technique to the network

Let us consider the problem size of the application network. If the reservoirs were discretized into 10 cells and the well pumps’ flow rates into five values (the constant speed pumps’ action space itself is 3,888) the size of the state space would be $10^8$ and the action space would be $4.86 \times 10^5$. This means that about $4.86 \times 10^{13}$ evaluations must be fulfilled in every timestep which is clearly computationally unfeasible. Using the splitting technique which explodes the system into the main distribution system and the well fields, the state space of the main distribution system is decreased to $10^5$ and the action space is reduced to 3,888.

Selecting the key reservoirs is a challenging task. After taking a look at Figure 1 one can observe that the waterworks of Sopron consists of two parts. Reservoirs 0–3 and pumps 0–2 gather the water from the wells while reservoirs 4–7 and pumps 4–7 supply the consumers with the water. Pump 3 connects these two parts and reservoirs 3 and 4 are on its suction and pressure sides, which suggests to select them as key reservoirs.

The above described method is highly network dependent. A more sophisticated approach for selecting the key reservoirs is the following. Each of the nine initial water level constellations was performed for each possible reservoir pair $(7 + 6 + 5 + 4 + 3 + 2 + 1 = 28)$. The objective values which correspond to the same initial level scenario were plotted in the same diagram. Each case showed that the selection of reservoirs 3 and 4 is reasonable. Figure 2 shows the average objective values of the nine cases.

RESULTS

As mentioned before, for test runs nine different test cases were defined which differ in the initial water volumes of the reservoirs; the objective function was the total cost (9). Each of the key reservoirs was discretized into 50 cells which was found to be a good compromise between the computer demand and the quality of the solutions. In the

![Figure 2](https://iwaponline.com/ws/article-pdf/13/5/1367/415131/1367.pdf)
case of the GA, the number of evaluations was set to $12 \times 10^6$ and $10^3$ single runs were performed for each test case. The NEOS solvers were used with their default settings. The ADP and the GA approach were implemented in C++ and run on a workstation equipped with Intel Core i7–2600 CPU @ 3.4 GHz CPU and 8 GB RAM, for the computations only one single core was used.

Comparing the results obtained by the different approaches

A single computation took 5 minutes for the GA, 1 minute for the ADP, and less than 1 second for the NEOS solvers. Note that the parameters of the NEOS Servers are unknown. Table 1 summarizes the results.

As the table shows, the objective values obtained by the ADP and the GA are nearly the same and 3.5% higher than the best of the NEOS solvers. The ADP has the advantage that the solution satisfies the whole constraint system in every single run while the GA fails in 4% of the cases. It is also clearly seen that there is no ‘superior’ solver which would have provided the best solution in every case.

Analysis of the results obtained by the ADP in the context of real life applications

The presented ADP method was set up on the server machine of the waterworks of Sopron for daily use. The initial state of the system (actual reservoir volumes, well flow rates) is read from a SCADA system and transformed to the file system of the optimization software. The optimization process is started by human operators, and the results can be directly performed on the system. A new optimization is automatically done if the forecasted reservoir levels (based on the previous optimization) are significantly different (more than 5%) from the measured ones.

Although the algorithm performs well, human operators found the switching number of the pumps still high, see Figure 3. After changing the objective function to the switching number another schedule was obtained which is depicted in Figure 4. The effect is obvious: in the first case the total cost was 7,186€ and the switching number was 72, while in the second case the values were 8,555€ and 29 switches. (Both belong to test case 1.) The linear combination of these quantities as the objective function is also possible and does not need any modification in the program code.

DISCUSSION

In this paper a novel approximate dynamic programming approach was introduced for solving water network optimization problems. The method is based on two key ideas. The optimization problem was exploded into the

<table>
<thead>
<tr>
<th>Test case</th>
<th>ADP</th>
<th>GA</th>
<th>Cbc</th>
<th>Glpk</th>
<th>Gurobi</th>
<th>MOSEK</th>
<th>scip</th>
<th>XPressMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7,186</td>
<td>7,164</td>
<td>7,171</td>
<td>7,090</td>
<td>6,990</td>
<td>7,334</td>
<td>7,028</td>
<td>6,987</td>
</tr>
<tr>
<td>2</td>
<td>7,243</td>
<td>7,207</td>
<td>7,001</td>
<td>7,007</td>
<td>6,975</td>
<td>7,250</td>
<td>7,153</td>
<td>7,086</td>
</tr>
<tr>
<td>3</td>
<td>7,226</td>
<td>7,150</td>
<td>6,919</td>
<td>6,961</td>
<td>6,862</td>
<td>6,934</td>
<td>6,935</td>
<td>6,859</td>
</tr>
<tr>
<td>4</td>
<td>7,147</td>
<td>7,209</td>
<td>7,066</td>
<td>6,991</td>
<td>6,948</td>
<td>6,898</td>
<td>6,920</td>
<td>6,930</td>
</tr>
<tr>
<td>5</td>
<td>7,386</td>
<td>7,375</td>
<td>7,058</td>
<td>–</td>
<td>7,379</td>
<td>7,614</td>
<td>7,280</td>
<td>7,007</td>
</tr>
<tr>
<td>6</td>
<td>7,295</td>
<td>7,301</td>
<td>7,011</td>
<td>–</td>
<td>6,959</td>
<td>7,382</td>
<td>7,300</td>
<td>7,378</td>
</tr>
<tr>
<td>7</td>
<td>7,131</td>
<td>7,220</td>
<td>7,071</td>
<td>–</td>
<td>6,889</td>
<td>7,018</td>
<td>6,959</td>
<td>6,929</td>
</tr>
<tr>
<td>8</td>
<td>7,252</td>
<td>7,161</td>
<td>7,085</td>
<td>6,942</td>
<td>6,928</td>
<td>7,103</td>
<td>7,217</td>
<td>6,905</td>
</tr>
<tr>
<td>9</td>
<td>7,169</td>
<td>7,183</td>
<td>7,210</td>
<td>7,083</td>
<td>7,005</td>
<td>7,043</td>
<td>7,057</td>
<td>7,303</td>
</tr>
<tr>
<td>Average</td>
<td>7,239</td>
<td>7,219</td>
<td>7,066</td>
<td>7,012</td>
<td>6,993</td>
<td>7,175</td>
<td>7,094</td>
<td>7,043</td>
</tr>
<tr>
<td>Success rate [%]</td>
<td>100</td>
<td>96</td>
<td>100</td>
<td>67</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
main distribution system and the well fields. Each well field turned into a separate LP problem which is easy to solve, moreover, the state space of the remaining system also decreased. This methodology works in the cases when the energy consumption of the wells is negligible (or linear) and their flow rate can be set continuously, which is a reasonable assumption in the case of the most realistic distribution system. The second idea was to reduce the state space size arbitrarily by defining the state space over a pair of ‘key reservoirs’. Although the new method uses significant approximations the obtained objective values have nearly the same quality as a specialized genetic algorithm (Selek et al. 2012) but they always fulfil all constraints. The method was also compared to six state-of-the-art optimization solvers and the objective values were only 3% worse which shows the potential of this research direction.

Although the presented mathematical model is based on a ‘flow-only’ hydraulic model, the ADP algorithm may be extended for water networks where coupled hydraulic simulation (i.e. solving the energy and mass equations) is needed. Description of the state space by using only the key reservoirs does not have any restriction on the hydraulic model. Splitting the system into subsystems can be more complex. However, the LP-subproblems still remain and a local controller can set up the appropriate revolution.

Figure 3 | Optimal pump schedule obtained by the ADP. Objective: total cost. The expensive tariff hours have a grey background colour.
number in order to achieve the desired flow rate computed by the LP solver. This is a big potential of the application area and it would be worth comparing the obtained results to general non-linear optimization solvers.

ACKNOWLEDGEMENTS

The presented work has been carried out within the OPUS project (Project ID: 138349) funded by the Academy of Finland. This work was partially supported by the scientific program of the ‘Development of quality-oriented and harmonized R+D+I strategy and functional model at BME’ project, New Hungary Development Plan (Project 15 ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

REFERENCES


NEOS Server 2013 http://neos-server.org/neos/.


First received 17 October 2012; accepted in revised form 20 March 2013. Available online 14 September 2013