

# Water network operational optimization: Utilizing symmetries in combinatorial problems by dynamic programming

József Gergely Bene / István Selek

Received 2011-06-27, revised 2011-11-21, accepted 2012-02-03

## Abstract

This paper introduces a dynamic programming (DP) approach for solving deterministic combinatorial operational optimization problem of water distribution networks. The implementation of dynamic programming over control domain using permutational symmetries is suggested to replace the state space based DP procedures. To enhance the understanding an application on a ub-network of the water supply and distribution network of the city of Sopron (Hungary) is presented which is sufficiently small to track the (pseudo) state space and approach related quantities.

## Keywords

dynamic programming · combinatorial operational optimization · water distribution systems

## Acknowledgement

The presented work has been carried out within Project OPUS (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 ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

The authors would like to express their gratitude to Enso Ikonen, László Kullmann and Csaba Hős for the personal communication.

## József Gergely Bene

Department of Hydrodynamic Systems, Budapest University of Technology and Economics, H-1521, Budapest, Hungary, P.o. Box 91., Hungary  
e-mail: bene@hds.bme.hu

## István Selek

Systems Engineering Laboratory, Department of Process and Environmental Engineering, University of Oulu, FIN-90014, University of Oulu, Finland, P.o.Box 4300., Finland  
e-mail: istvan.selek@artificialevolution.net

## Nomenclature

Notation	Description	
$N_{\text{pump}}$	Number of pumps in the main distribution network	
$N_{\text{res}}$	Number of water reservoirs in the main distribution network	
$N_{\text{pow}}$	Number of power stations in the main distribution network	
$N_{\text{well}}$	Number of well fields supplying the main distribution network	
Variable	Unit	Description
$q_{i,t}^{\text{pump}}$	$m^3/h$	Flow rate of pump $i$ at time $t$
$q_{i,t}^{\text{well}}$	$m^3/h$	Flow rate of well $i$ at time $t$
$d_{i,t}$	$m^3$	$i$ th water demand between time $t$ and $t+1$
$P_{i,t}^{\text{pump}}$	kW	Power consumption of pump $i$ at time $t$
$V_{i,t}$	$m^3$	Water volume of reservoir $i$ at time $t$
$E_{i,t}$	kWh	Supplied energy by power station $i$ at time $t$
$w_t$	€/kWh	Energy tariff at time $t$

## 1 Introduction

Optimization is a highly researched topic energized by the industry. In waterwork systems the need for optimization is twofold: it is required either at design stage of waterworks [7, 8, 16, 34] or more frequently the demand focuses towards operational level: having a given waterwork topology one aims to achieve an optimal control of the active hydraulic elements (pumps, valves) satisfying water demand with minimal energy consumption. Sophisticated operation can result in significant savings even in small scale waterworks.

So far the scientific research of this topic resulted in an extensive growth of the optimization techniques. Currently metaheuristics dominate the research arena [1, 5, 6, 31, 32], they are widely applied due to their flexibility and robustness. However, they often provide poor solutions if the CPU time is limited for optimization making them unsuitable for online control [23].

Besides heuristics, deterministic solvers own a great fraction of the stake. Among those, dynamic programming (DP) has long been recognized as powerful tool and global optimizer,

however it targets only continuous optimal control problems in the field of operational optimization of waterwork systems [9–15, 35]. In the context of dynamic programming discrete pump models are rarely used, the literature lacks the application of DP on combinatorial problems introduced by on/off type pumps or/and valves implemented in a water distribution system.

To decrease this research gap here the utilization of dynamic programming on combinatorial problems related to operational optimization of water distribution networks is examined. In order to achieve better understanding a small scale water network is used as illustration.

The paper is organized as follows: Section 2 describes the dynamics of the investigated water networks while Section 3 gives a brief overview on dynamic programming approaches on the context of water resources management problems. Section 4 introduces a novel dynamic programming approach as the main contribution of this paper. Sections 5 and 6 present an application example of the introduced technique where a combinatorial and a mixed-combinatorial problem are under inspection. Finally, Section 7 includes the conclusions and hints the future work.

## 2 The water distribution system model

The primary goal of a water distribution system is to satisfy the residential and industrial demand by delivering water from sources (wells) to users. A simple water supply system is shown in Fig. 1. The network consists of water reservoirs, water demands, power station, water source (well field) and pumping stations transferring the water through the distribution system. In essence a pumping station (represented by single "Pump" units in the model) is a group of individual (fixed or variable speed) pumps running parallel. As the terminology suggests, fixed speed pumps can be operated only at a given revolution number, thus they are often termed as on/off type control elements. In contrast, variable speed pumps use frequency converter to allow the setting of arbitrary speed within a given range achieving higher degree of operational freedom.

To derive the water distribution network model the following assumptions were taken into account:

- The fluid (water) is incompressible.
- The water demands are deterministic, and known a priori.
- Only constant speed pumps are operated in the main distribution network which is connected with water source by variable speed pumps (considered as a typical water distribution system configuration).
- The variations of the pump operation points at a given pump speed are negligible or in other words the flow rates and corresponding power consumptions are correlated. Due to the authors' best knowledge most of the real water networks can be modeled in this manner.

The former condition allows the (fixed speed or variable speed) pumping stations to be represented by a (finite or infinite) set of flow rates and corresponding energy consumptions. Using this, the term "pumping station" is often referred as "pump" throughout this paper, however under both phrases one means a set of individual pumps.

### 2.1 System dynamics

The dynamics of a water distribution system obeys the *law of continuity* which represents mass balance of the nodes of the network. Using continuity the water volume of a reservoir ( $V_{i,t}$  [ $m^3$ ]) is determined by the inflows (incoming pump flow:  $q_{j,t}^{\text{pump}}$  [ $m^3/h$ ], well pump flow:  $q_{j,t}^{\text{well}}$  [ $m^3/h$ ]) and the outflows (outgoing pump flow:  $q_{j,t}^{\text{pump}}$  [ $m^3$ ] and water demand:  $d_{j,t}$  [ $m^3$ ]) at time  $t$ :

$$V_{i,t+1} = V_{i,t} + \left( \sum_{j \in C_r^{p+}} q_{j,t}^{\text{pump}} + \sum_{j \in C_r^w} q_{j,t}^{\text{well}} - \sum_{j \in C_r^{p-}} q_{j,t}^{\text{pump}} \right) \Delta t - \sum_{j \in C_r^d} d_{j,t}, \quad (1)$$

$i = \{0, 1, \dots, N_{\text{res}} - 1\}$  and  $t = \{0, 1, \dots, T - 1\}$  and  $C_r^* = C_r^*(i)$ , where  $\star = \{p+, w, p-, d\}$  describe the set of hydraulic elements (incoming pumps, wells, outgoing pumps and water demands, respectively) connected to the same node.

In operational aspects the energy supply of the pumping stations plays a key role. The supply topology usually exhibits a one to more correspondence, that is, a power station typically feeds a group of pumping stations (typically 2,3). The supplied *electric energy* ( $E_{i,t}$  [kWh]) is then expressed as the sum of the individual power consumptions during time period  $t$

$$E_{i,t} = \sum_{j \in C_s^p} p_{j,t}^{\text{pump}} \Delta t, \quad i = \{0, 1, \dots, N_{\text{pow}} - 1\} \quad (2)$$

while the consumed power of a particular pumping station is a function of the established flow rates:

$$p_{j,t}^{\text{pump}} = f(q_{j,t}^{\text{pump}}), \quad j = \{0, 1, \dots, N_{\text{pump}} - 1\}. \quad (3)$$

### 2.2 Constraint system

The operation of a water distribution network is limited by the constraint system which constitutes water reservoir boundaries, energy consumption limits and pump operational restrictions. The non-negativity of all model variables is assumed, however, the method to be presented in the following sections does not require this restriction.

By having a finite reservoir capacity, the stored water must not exceed the upper and lower reservoir boundaries on the operational horizon expressed as a general time dependent requirement:

$$V_{i,t}^{\min} \leq V_{i,t} \leq V_{i,t}^{\max} \quad (4)$$

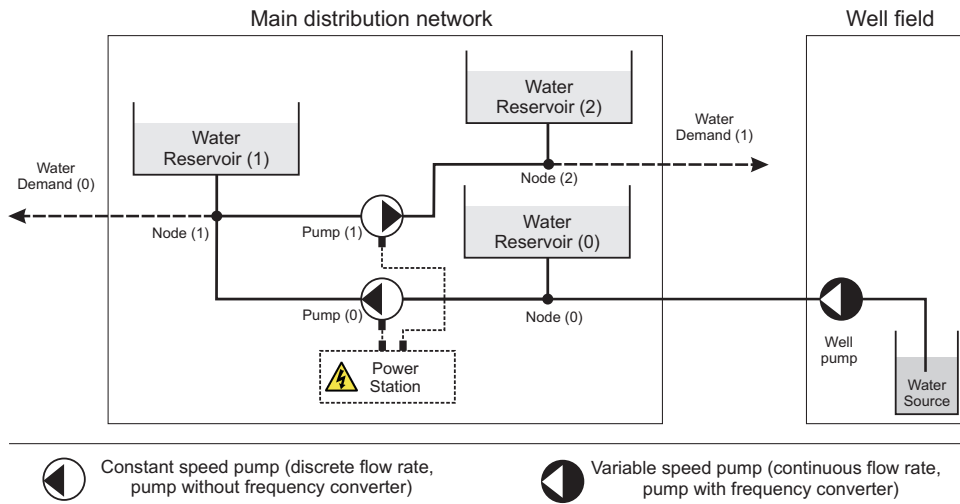


Fig. 1. Network topology model

On the other hand power station usage constraints

$$E_{i,t} \leq E_{i,t}^{\max}, \quad (5)$$

represent an upper limit on the supplied energy at each time step.

The water exploitation from wells must be 'smooth'. As an industrial requirement this condition restricts the state of the well pump to be changed at each time step on the optimization horizon. The switch is possible only at certain time instants and the pump must keep the established flow rate for a fixed time period ( $t \in T^{\text{fix}}$ ):

$$q_{j,t}^{\text{well}} = q_{j,t-1}^{\text{well}}, \quad j \in \{0, 1, \dots, N_{\text{well}} - 1\} \quad (6)$$

### 2.3 Objective function

Let  $w_t$  the price of the consumed energy [€/kWh] at time  $t$  which typically incorporates 'cheap' and 'expensive' pricing periods on the optimization time scale. The total operational cost ( $TC$ ) is related to the pump energy consumption as:

$$TC = \sum_{t=0}^{T-1} \left( w_t \sum_{i=0}^{N_{\text{pow}}-1} E_{i,t} \right). \quad (7)$$

By putting everything together, an optimal control problem is formulated: determine a suitable pump action on the optimization time horizon which minimizes (7) subject to constraints (4) – (6) satisfying the water demands (1).

### 2.4 The investigated network

As test-network a small scale sub-system of the water distribution network of Sopron (Hungary) was used (see Fig. 1). By implementing this small network the obtained optimal control problem can be efficiently solved although its search space is still too large for complete enumeration (Section 5.1 and [1]). There are dozens of deterministic/stochastic approaches which can get close to the global optimum (or determine the global optimum) within seconds. For informative comparison some of those were executed on the test network (results are summarized by Sections 5.2 and 6.2).

Tab. 1. Flow rates and consumed powers of the constant speed pumps

(state)	$q_0$ [m <sup>3</sup> /h]	$p_0$ [kW]	$q_1$ [m <sup>3</sup> /h]	$p_1$ [kW]
0	0	0	0	0
1	100	55	320	110
2	420	110	550	220

Tab. 2. Minimum and maximum limits of the reservoirs. (The values with \* were changed for the different optimization tasks.)

$t$ (period)	$V_{0,t}^{\min}$ [m <sup>3</sup> ]	$V_{0,t}^{\max}$ [m <sup>3</sup> ]	$V_{1,t}^{\min}$ [m <sup>3</sup> ]	$V_{1,t}^{\max}$ [m <sup>3</sup> ]	$V_{2,t}^{\min}$ [m <sup>3</sup> ]	$V_{2,t}^{\max}$ [m <sup>3</sup> ]
0	1700	1700	200	200	1800	1800
1-23	*100	2000	100	3600	100	2000
24	*1600	1800	100	300	1700	1900

The motivation behind this example is not to solve the test network and announce the superiority of our method, but to solve it with understanding. This system is small enough to track the state space and approach related quantities which would not be possible otherwise.

The operational data of the test network is summarized by the following tables. Table 1 shows the set of possible flow rates and corresponding power consumptions of pumping stations located within the main distribution network. Both pumping stations are built up by constant speed pumps giving discrete control action sets.

As optimal control problem the one-day (24h) optimal pumping policy of the test network is investigated on hourly basis ( $T = 24, \Delta t = 1h$ ). Using this discretization the reservoir constraints (upper and lower capacities) are highlighted in each term by Table 2 (term  $t = 0$  represents the initial conditions) while Table 3 shows the user demand and energy tariff. Finally, the operational constraint on power station restricts the energy supply which can not exceed  $E_{0,t}^{\max} = 300$  [kWh] on the optimization time horizon.

The handling of the water feed from well field to main distri-

bution network is twofold. At first, the water flow of the well pump is assumed to be known priori ( $q_{0,t}^{\text{well}} = 330 \text{ [m}^3/\text{h]}$ ) resulting a combinatorial optimization problem where only discrete decisions are available as possible pump actions. This problem is used as a reference, for comparison see [1] and [2]. Motivated by the industry a mixed-combinatorial problem is introduced in the second case where (besides discrete delivery pumps) the flow rate of the well pump is considered as control variable as well having continuous decision space.

### 3 Dynamic programming in water resources management problems

Using the continuity equation (1) the dynamics of the system can be represented in state space form:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta t \mathbf{B}\mathbf{u}_t + \mathbf{D}\mathbf{d}_t \quad (8)$$

where reservoir water volumes form the state vector  $\mathbf{x}_t = (V_{0,t}, \dots, V_{N_{\text{res}}-1,t})^T [\text{m}^3]$  while control variable  $\mathbf{u}_t = (q_{0,t}^{\text{pump}}, \dots, q_{N_{\text{pump}}-1,t}^{\text{pump}})^T [\text{m}^3/\text{h}]$  represents the pump flow rate decisions and  $\mathbf{d}_t [\text{m}^3]$  denotes the deterministic water inflows/demands between time  $t$  and  $t + 1$ . Using the principle of optimality [20] Bellman and Dreyfus introduced Discrete Dynamic Programming (DDP) which requires quantized state space to recursively compute the iterative functional equation (*cost-to-go* function)

$$J_t(\mathbf{x}_t) = \min_{\mathbf{u}_t} \left\{ c_t(\mathbf{u}_t, \mathbf{x}_t) + J_{t+1}(\mathbf{x}_{t+1}) \right\} \quad (9)$$

to obtain the optimal control sequence  $\{\mathbf{u}_t^*\}_{t=0}^{T-1} = \{\mathbf{u}_0^*, \dots, \mathbf{u}_{T-1}^*\}$  which minimizes (7) with respect to the constraint system (constraint handling is usually incorporated within the transition cost  $c_t(\mathbf{u}_t, \mathbf{x}_t)$  by penalty functions). The need of discretization of the essentially continuous state space makes this approach insufficient on water network operational optimization problems having discrete control elements (e.g. on/off type pumps and/or valves) as follows.

#### 3.1 DDP Difficulties

In general DDP implements the 'pulling' model [33] which requires the state space  $X$  to be quantized beforehand, that is, the state nodes must be generated and stored prior to initiating the computations. When no information is available about the form of the cost-to-go function  $J_t(\mathbf{x}_t)$ , each state variable is uniformly quantized [18]. However, there are dozens of more efficient discretization methods available in the literature [25–28, 30] which may even resolve the curse of dimensionality (the exponential growth of the number of nodes nodes in  $X$  subject to state space dimension) on specific problem classes [29].

Due to quantization the value of the cost-to-go function is calculated over a finite set of node points which requires inverted state dynamics to obtain controls  $\mathbf{u}_t$  as a function of consecutive states  $\mathbf{x}_{t+1}, \mathbf{x}_t$ . However in deterministic problems if the space

of the possible decisions is discrete mostly no control action exists for state pairs  $\{\mathbf{x}_t, \mathbf{x}_{t+1}\}$  on a finite grid even for water systems with simple invertible dynamics.

To overcome the difficulties one may use a similar approach to (Linear Programming) LP-relaxation which is a widely applied solution technique in integer programming [21]. Relaxation allows the decision variables to have any fractional value on a given continuous set (usually on the interval  $[0, 1]$ ). First an optimal solution is generated allowing each decision variable to be relaxed. Then the relaxed values are systematically eliminated from the suboptimal solution by recursively solving subproblems using linear programming methods (LP). This procedure was introduced by [22] and it is well known as *branch and bound* or *tree search* implementing a key idea of enumerating feasible solutions "around" the relaxed optimal trajectory such that the optimal integer solution is found.

In the interpretation, the linear programming must be replaced by dynamic programming to ensure the finding of global optimum at each stage. The high number of subproblems to be solved until the feasible solution is found makes this approach cumbersome to employ for practical problems. unpractical.

Other possibility is the use of DDP by parceling the state space i.e. introducing state cells by partitioning instead of state nodes by sampling. At this point the problem can be handled as Markov Decision Process (MDP) [2] utilizing the distribution of states over partitions. Although the evolution of the system is deterministic the transition between partitions is assigned by probabilities: from one state cell more possible target cells can be reached by a given control action depending on the position of the initial state of the system  $\mathbf{x}_t$  within the state cell. The transition probability matrix is simple to compute (e.g. by Monte Carlo simulation), however it may consume remarkable CPU time even for simple systems depending on partition density. The resulting MDP problem is then solved by stochastic dynamic programming and the obtained solution is back-tested on the original deterministic problem.

On the other hand, partitions can be simply treated as sink cells by keeping only the best candidate solution within a cell at time  $t$  (all others are removed causing remarkable information loss) [24]. Besides that, the method is simple to implement however, it usually obtains solution far from global optimum when coarse discretization is used. Like MDP approach, the finding of global optima is guaranteed by a sufficiently dense partitioning (number of partitions approaches to infinity) in which case the problems grow far beyond the capabilities of digital hardware.

Another alternative to resolve the state space discretization problem is the forward generation of the state nodes on the optimization time horizon simply by using the state transition function (8) and feasible decision sequences. The method only generates states that can be reached from a given initial state (or set of initial states) [33]. The state generation as well as dynamic programming algorithm (cost-to-go recursion) involve substantial computation therefore their sequential application is

**Tab. 3.** Water demand data and energy tariff

	$t$	0	1	2	3	4	5	6	7	8	9	10	11
$d_{0,t}$	$[m^3]$	35	15	14	24	29	39	53	65	65	55	52	53
$d_{1,t}$	$[m^3]$	196	81	76	133	165	216	296	362	363	308	292	295
$w_t$	$[\text{€}/\text{kWh}]$	1	1	1	1	1	1	1	1	2	2	2	2
	$t$	12	13	14	15	16	17	18	19	20	21	22	23
$d_{0,t}$	$[m^3]$	61	60	53	53	57	59	60	72	71	69	52	34
$d_{1,t}$	$[m^3]$	343	334	298	296	319	330	336	403	398	384	288	189
$w_t$	$[\text{€}/\text{kWh}]$	2	2	1	1	1	1	2	2	2	1	1	1

computationally inefficient.

To overcome the computational difficulties a new dynamic programming approach is introduced.

#### 4 Dynamic programming in control domain

The essential idea the proposed technique replaces the original state space and implements discrete dynamic programming in a series of pseudo states considering symmetries. The pseudo state variable is defined over the control domain where symmetries are introduced by the invariance of the original state  $\mathbf{x}$  to the permutations of a control sequence. We call this concept as *permutational invariance*.

The permutational invariance of a system represented by a general nonlinear dynamics

$$\mathbf{x}_{t+1} = \mathbf{f}_t(\mathbf{x}_t, \mathbf{u}_t), \quad t = 0, \dots, T - 1 \quad (10)$$

where  $\mathbf{x}_t = (x_{1,t}, \dots, x_{n,t})^T$  is the state vector and  $\mathbf{u}_t = (u_{1,t}, \dots, u_{m,t})^T$  denotes the control vector is defined as follows.

Let  $\Gamma_{j,t}$  denote the set of all possible permutations of the control sequence (the  $j$ th component of the control vector)  $\{u_{j,\tau}\}_{\tau=0}^t = \{u_{j,0}, u_{j,1}, \dots, u_{j,t}\}$  by time  $t$  and let  $\Upsilon_t = \Gamma_{1,t} \times \Gamma_{2,t} \times \dots \times \Gamma_{m,t}$  be the set of all possible control actions over the permutation sub-sets by time  $t$ , that is,

$$\Upsilon_t = \left\{ \left( \{u_{1,\tau}\}_{\tau=0}^t, \dots, \{u_{m,\tau}\}_{\tau=0}^t \right) \mid \{u_{j,\tau}\}_{\tau=0}^t \in \Gamma_{j,t}, j = 1, \dots, m \right\}. \quad (11)$$

By determining all possible permutation sets, the decision space becomes partitioned:  $D_t = \bigcup_{j=1}^{N_{p,t}} \Upsilon_{j,t}$  where  $D_t$  is the set of all possible control sequences by time  $t$  and  $N_{p,t}$  denotes the number of permutation sets at time  $t$ . Applying the controls  $\{\mathbf{u}_\tau^{(1)}\}_{\tau=0}^t$  and  $\{\mathbf{u}_\tau^{(2)}\}_{\tau=0}^t$  for a system represented by (10) let  $\mathbf{x}_{t+1}^{(1)}$  be the obtained state ( $\mathbf{x}_{t+1}^{(2)}$  respectively). The underlying system is called permutationally invariant if

$$\mathbf{x}_{t+1}^{(1)} = \mathbf{x}_{t+1}^{(2)}, \quad (12)$$

for every  $\{\mathbf{u}_\tau^{(1)}\}_{\tau=0}^t, \{\mathbf{u}_\tau^{(2)}\}_{\tau=0}^t \in \Upsilon_{j,t}$  pairs ( $\forall j \in \{1, \dots, N_{p,t}\}$ ) and  $t \in \{0, 1, \dots, T - 1\}$ . In other words, control actions which are members of the permutation set  $\Upsilon_{j,t}$  map the system into the same state.

By using the integrator form of the system dynamics (8)

$$\mathbf{x}_{t+1} = \mathbf{x}_0 + \sum_{\tau=0}^t \Delta t \mathbf{B} \mathbf{u}_\tau + \sum_{\tau=0}^t \mathbf{D} \mathbf{d}_\tau \quad (13)$$

the commutative operations applied to the control vector sequence highlights the permutational invariance. In other words, the water volume of a reservoir at time  $t$  depends on the total delivered water (cumulative defined using the control variable) rather than the schedule itself.

Using permutational symmetries let us define the pseudo state

$$\boldsymbol{\xi}_t = \sum_{\tau=0}^t \Delta t \mathbf{u}_\tau \quad (14)$$

denoting the total delivered water by time  $t$ . By recursively computing the iterative functional equation using pseudo states

$$\hat{J}_{t+1}(\boldsymbol{\xi}_{t+1}) = \min_{\mathbf{u}_t} \left\{ \hat{c}_t(\mathbf{u}_t, \boldsymbol{\xi}_t) + \hat{J}_t(\boldsymbol{\xi}_t) \right\} \quad (15)$$

the above defined water network optimal control problem can be solved achieving the global optimum. Here the forward iteration of (15) is preferred due to the setup of the optimization problem: besides the initial state  $\mathbf{x}_0$  at the end of the optimization period usually a set of the achievable states (target set) is given instead of a particular target state  $\mathbf{x}_T$ .

By substituting the pseudo state into (13) we obtain the dynamics

$$\mathbf{x}_{t+1} = \mathbf{x}_0 + \mathbf{B} \boldsymbol{\xi}_t + \sum_{\tau=0}^t \mathbf{D} \mathbf{d}_\tau \quad (16)$$

connecting the domains of the original  $\mathbf{x} \in X$  and pseudo states  $\boldsymbol{\xi} \in \Phi$ . In combinatorial problems  $\Phi$  is essentially discrete where the nodes are defined by the permutations of the control sequences. For the recursion of (15) the 'reaching' dynamic programming model [33] is applied, which generates the states and determines the optimal decisions simultaneously. As great benefit the handling of permutations and the forward recursion can be simply managed by using basic operations (insert, find, compare, etc.) on arrays.

Besides the simple implementation the introduced approach solves the discretization problem of the state space as well. Through dynamics (16) the discrete pseudo state space simultaneously introduces a grid on  $X$  as well. The nodes are introduced only on the achievable subset of the original state space at time  $t$  by a non uniform discretization. In what follows computer resources are not wasted by introducing nodes on  $X$  beforehand which can not be reached at time  $t$ . For each introduced node on state space there is a corresponding control action which can be observed as a straightforward corollary of the definition of the pseudo state variable.

#### 4.1 Extended reservoir constraint system

Using problem specific information further reduction on the size of the search space of the optimization problem can be achieved.

Reservoir volume constraints are usually stricter at the end of the optimization period ( $t = 24$ ) than before ( $t < 24$ ) resulting a 'sudden jump' in constraint system (see Fig. 2). This produces a set of states – called *dead storage* – from which the final constraint can not be satisfied by any possible pump action(s). By implementing smooth reservoir constraints in time, the candidate solutions are not allowed to evolve into dead storage spaces. This ensures the avoiding of the creation of such state nodes from which it is not possible to reach the target set. The following equations were used to find a time dependent corridor as extended constraint system for reservoirs:

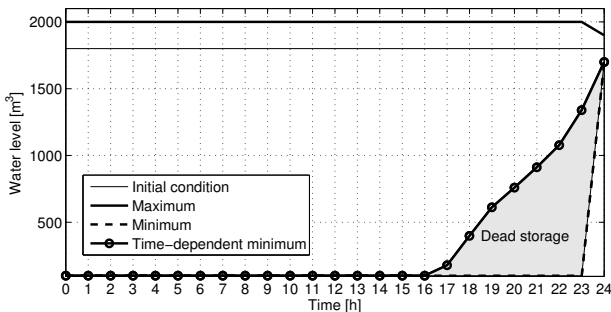


Fig. 2. Dead storage of Water Reservoir(2)

$$\begin{aligned}
 V_{0,t}^{\min} &= \max(100, V_{0,t+1}^{\min} + 330) \\
 V_{0,t}^{\max} &= \min(2000, V_{0,t+1}^{\max} + 330 + 420) \\
 V_{1,t}^{\min} &= \max(100, V_{1,t+1}^{\min} + d_{0,t} - 420) \\
 V_{1,t}^{\max} &= \min(3600, V_{1,t+1}^{\max} + d_{0,t} + 550) \\
 V_{2,t}^{\min} &= \max(100, V_{2,t+1}^{\min} + d_{1,t} - 550) \\
 V_{2,t}^{\max} &= \min(2000, V_{2,t+1}^{\max} + d_{1,t})
 \end{aligned} \quad (17)$$

which must be computed backward ( $t=23, \dots, 1$ ). For example Fig. 2 shows the extended constraint system for Water Reservoir (2).

#### 5 Water network optimization as combinatorial problem

As indicated in Section 2.4, first, the optimal pump schedule of the reference network with fixed well inflow is investigated. The system evolves according to

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \mathbf{u}_t + \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \mathbf{d}_t \quad (18)$$

where  $\mathbf{x}_0 = (1700, 200, 1800)^T$ ,  $\mathbf{d}_t = (330, d_{0,t}, d_{1,t})^T$  and

$$\mathbf{u}_t = \begin{pmatrix} u_{0,t} \in \{0, 110, 420\} \\ u_{1,t} \in \{0, 320, 550\} \end{pmatrix} \quad (19)$$

The goal is to minimize the total cost of the operation

$$\begin{aligned}
 TC = \sum_{t=0}^{23} & \left[ (6.40 \cdot 10^{-1})u_{0,t} - (9.00 \cdot 10^{-4})u_{0,t}^2 \right. \\
 & \left. + (3.16 \cdot 10^{-1})u_{1,t} + (8.65 \cdot 10^{-5})u_{1,t}^2 \right] w_t
 \end{aligned} \quad (20)$$

subject to reservoir constraints  $\mathbf{x}_t^{\min} \leq \mathbf{x}_t \leq \mathbf{x}_t^{\max}$  and energy supply limitation:

$$\begin{aligned}
 (6.40 \cdot 10^{-1})u_{0,t} - (9.00 \cdot 10^{-4})u_{0,t}^2 + (3.16 \cdot 10^{-1})u_{1,t} \\
 + (8.65 \cdot 10^{-5})u_{1,t}^2 \leq 300
 \end{aligned} \quad (21)$$

Functions (20) and (21) were derived using second order interpolation on corresponding data pairs summarized by Table 1.

The obtained optimal control problem has two discrete decision variables giving 9 possible control actions at each time step. By going further in time the number of solution candidates explodes far beyond the capabilities of brute-force search techniques due to exponential growth.

#### 5.1 Magnitude of the search space

Although the topology of the test network is conceptionally simple the problem cannot be solved by exhaustive (brute-force) search. Indeed the size of the search space (number of possible solutions) equals  $(3^2)^{24} \approx 10^{23}$  (2 pumps, 3 possible states, 24 terms). Let us make an estimation on the total computational time required by complete enumeration of the search space. Making an irrational assumption that  $10^{15}$  candidate solutions can be evaluated in one second then the total time required for finding the optimum is still 2.529 years. However,  $10^{15}$  candidate solutions in one second needs approximately  $10^5$  teraflop (floating point operations per second), while the world's fastest supercomputer's (Tianhe-1A) performance is  $2.5 \times 10^3$  teraflop [4].

#### 5.2 Results

Using the introduced dynamic programming technique the size of the problem is highly scaled down. Fig. 3 depicts the exponential growth of the problem size (number of possible solutions) in time (appears as linear function on the logarithmic scale). Great reduction is achieved by exploiting permutational symmetries resulting 18 magnitude of order at the end of the optimization period. The constraint system further decreases the size of search space, obtaining it in enumerable size at 24th time step guaranteeing the finding of the global optimum.

In contrast to the original constraint system the relaxed reservoir constraints appear where the dead storage forms (usually at the end of the optimization horizon) obtaining additional reduction (about 1–2 magnitude of order) which is a significant achievement even in this simple case. Note that the size of the dead storage depends on the initial condition.

On the other hand, the size of the pseudo state  $\Phi$  grows in time unlike the state space  $X$ . For comparison the state space was

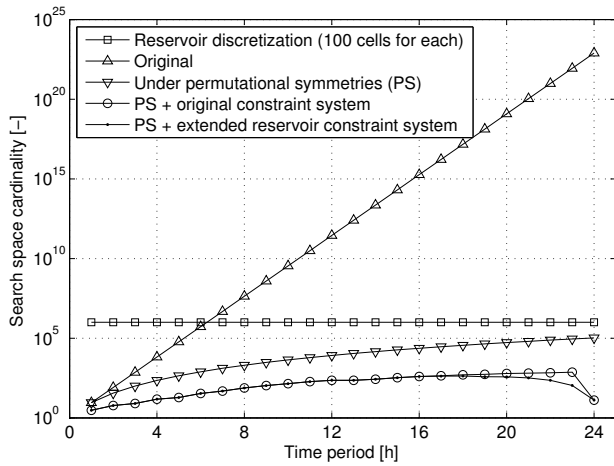


Fig. 3. Number of states for different 'discretization' methods

uniformly quantized using 100 cells on each coordinate considered as a sufficient discretization scale required to obtain satisfactory results by "conventional" dynamic programming.

The introduced method was implemented under C++ and executed on a computer equipped with Intel Core 2 Duo T6600 CPU (2.2 GHz) without parallelization. The achieved optimal cost was 5830 € which is obviously better than the best solution found by the genetic algorithm (5865 €, [1]) or by a conventional DP approach (6170 €, [2]). The optimization process consumed about 0.02-0.03 second CPU time, the corresponding optimal schedule is shown in Fig. 4.

The implementation of the reservoir constraints in pseudo state space is accomplished by using the integrator form (13) and rearranging the system equation (18)

$$\begin{aligned} \left( \mathbf{x}_{t+1}^{\min} - \sum_{\tau=0}^t \mathbf{D}\mathbf{d}_{\tau} - \mathbf{x}_0 \right) &\leq \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \boldsymbol{\xi}_t \\ &\leq \left( \mathbf{x}_{t+1}^{\max} - \sum_{\tau=0}^t \mathbf{D}\mathbf{d}_{\tau} - \mathbf{x}_0 \right), \end{aligned} \quad (22)$$

where the left and right hand side of (22) can be pre-calculated at the beginning of the optimization process.

In this particular example, the reservoir constraint system represents a 'cube' in the 3 dimensional state space  $X$  which is transformed into a (general) polygon lying within the 2 dimensional pseudo (state) space  $\Phi$ . This polygon is formed by a rectangle (min and max limits of Reservoirs 0 and 2 which are connected only with one pump) whose corners are cut off by the constraints on Reservoir 1 appear as 45 degree rotated cutting edges in  $\Phi$ , see Figs. 5 and 6. This particular reservoir is affected by two pumps (incoming and outgoing): the signed sum of the delivered water by pump 0 and 1 located within the main distribution network.

On the other hand the power station constraint handling is managed simply by eliminating the candidate solution form inventory which does not satisfy (21). Fig. 5 shows the optimal trajectory and constraint system over pseudo state space.

Using a constraint system 'slice' in pseudo space at time period 23 the shape of the cost-to-go function is shown in Fig. 6. It is interesting to note that the cost-to-go function is rugged (regardless to energy price peaks) although the objective function (20) implements a smooth, quadratic formula. Ruggedness appears due to the combinatorial nature of the problem: the finite control set.

Finally a comparison is given in Table 4 summarizing the obtained results by different optimization approaches. The table targets to give information on how the solvers respond to the constraint system. The investigation serves only informative purposes; all tested solvers were used with their default setup-parameters. As the capacity of a water reservoir is becoming narrower the number of the feasible solutions approaches zero. Stricter constraint system enhances the performance of DP (less nodes to be considered) while introduces challenges for other solvers. To implement this, three different optimization problems were introduced beyond the original task by decreasing the useful capacity of Reservoir 0. These setups were then solved by world-leader general purpose optimization methods as well, which are freely available on the internet at NEOS' homepage [3]. The consumed time was in the same order of magnitude as in the case of the authors' solver. The source code of NEOS model files (implemented under GAMS) and DP approach can be downloaded from [17].

## 6 Water network optimization as mixed-combinatorial problem

Motivated by the industry, here the flow rate of the well pump is considered as control variable as well having continuous decision space. The dynamics of the systems can be written

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \mathbf{u}_t + \begin{pmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \mathbf{d}_t \quad (23)$$

where  $\mathbf{x}_0 = (1700, 200, 1800)^T$ ,  $\mathbf{d}_t = (d_{1,t}, d_{2,t})^T$  and

$$\mathbf{u}_t = \begin{pmatrix} u_{0,t} \in \{0, 110, 420\} \\ u_{1,t} \in \{0, 320, 550\} \\ u_{2,t} \in [0, 500] \end{pmatrix}. \quad (24)$$

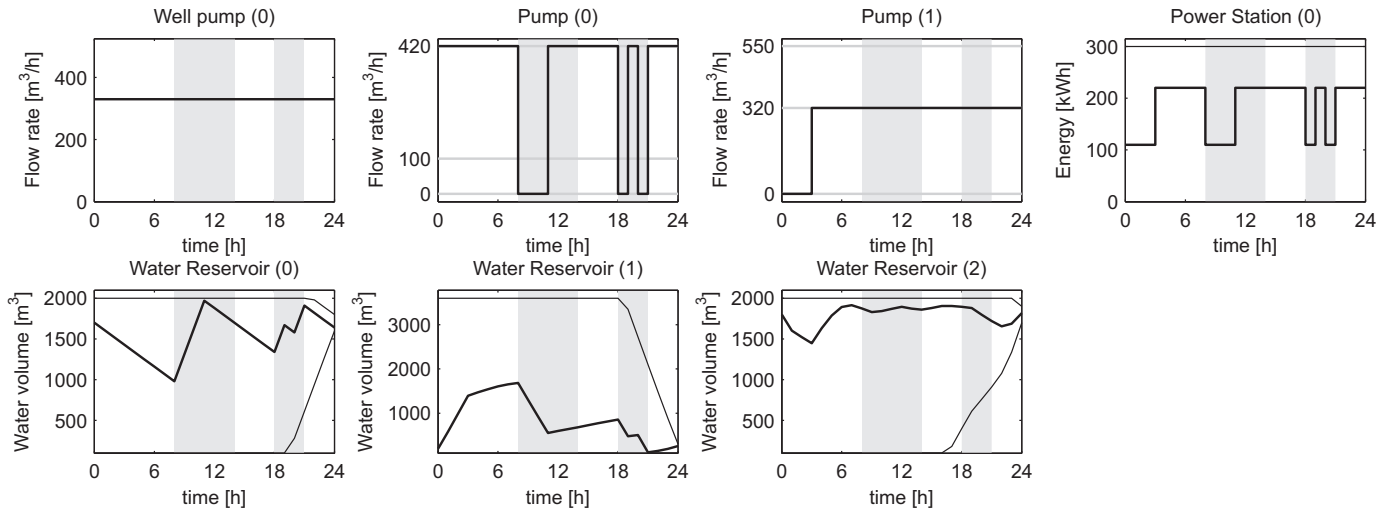
The water delivery of the well pump is restricted by constraint (6) on the optimization horizon. In what follows the control sequence  $\{u_{2,t}\}_{t=0}^{23}$  must have the following pattern (the flow rates are adjusted for billing periods):

$$\begin{aligned} \{u_{2,t}\}_{t=0}^{23} &= \{ \underbrace{330, 330, 330, 330, 330, 330, 330, 330,}_{q_1, q_1, q_1, q_1, q_1, q_1, q_2, q_2, q_2, q_2,} \\ &\quad \underbrace{q_3, q_3, q_3, q_4, q_4, q_4} \}. \end{aligned} \quad (25)$$

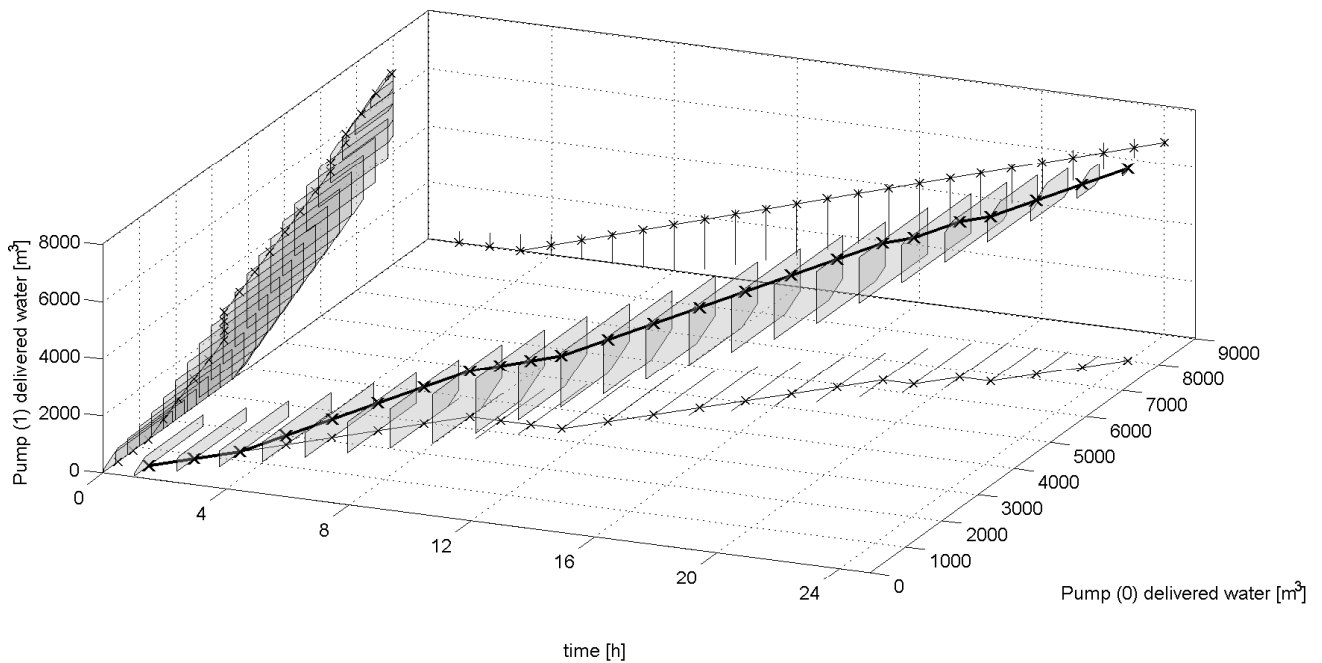
where the flow rates  $q_1, \dots, q_4$  can take any value on the interval  $[0, 500]$ . The corresponding objective function and constraint system were earlier defined in Section 5.

**Tab. 4.** Comparison of the optimal solutions found by the NEOS solvers and the proposed DP technique on different test problems (nf. = feasible solution not found).

$V_{0,1..23}^{min}$ [m <sup>3</sup> ]	$V_{0,24}^{min}$ [m <sup>3</sup> ]	Cost, NEOS solvers [€]						Cost [€] DP
		Cbc	Glpk	Gurobi	MOSEK	scip	XpressMP	
100	1600	<b>5830</b>	5920	6215	5940	5975	<b>5830</b>	<b>5830</b>
1000	1600	5940	6005	5940	6120	5975	<b>5920</b>	<b>5920</b>
1600	1600	6370	6170	6370	6395	6135	6140	<b>6115</b>
1700	1700	nf.	nf.	nf.	nf.	nf.	nf.	nf.



**Fig. 4.** Optimal schedule: 5830 €. Peak charging periods (2 €/kWh) are gray shaded while off-peak periods (1 €/kWh) are colourless. Thin lines represent constraints.

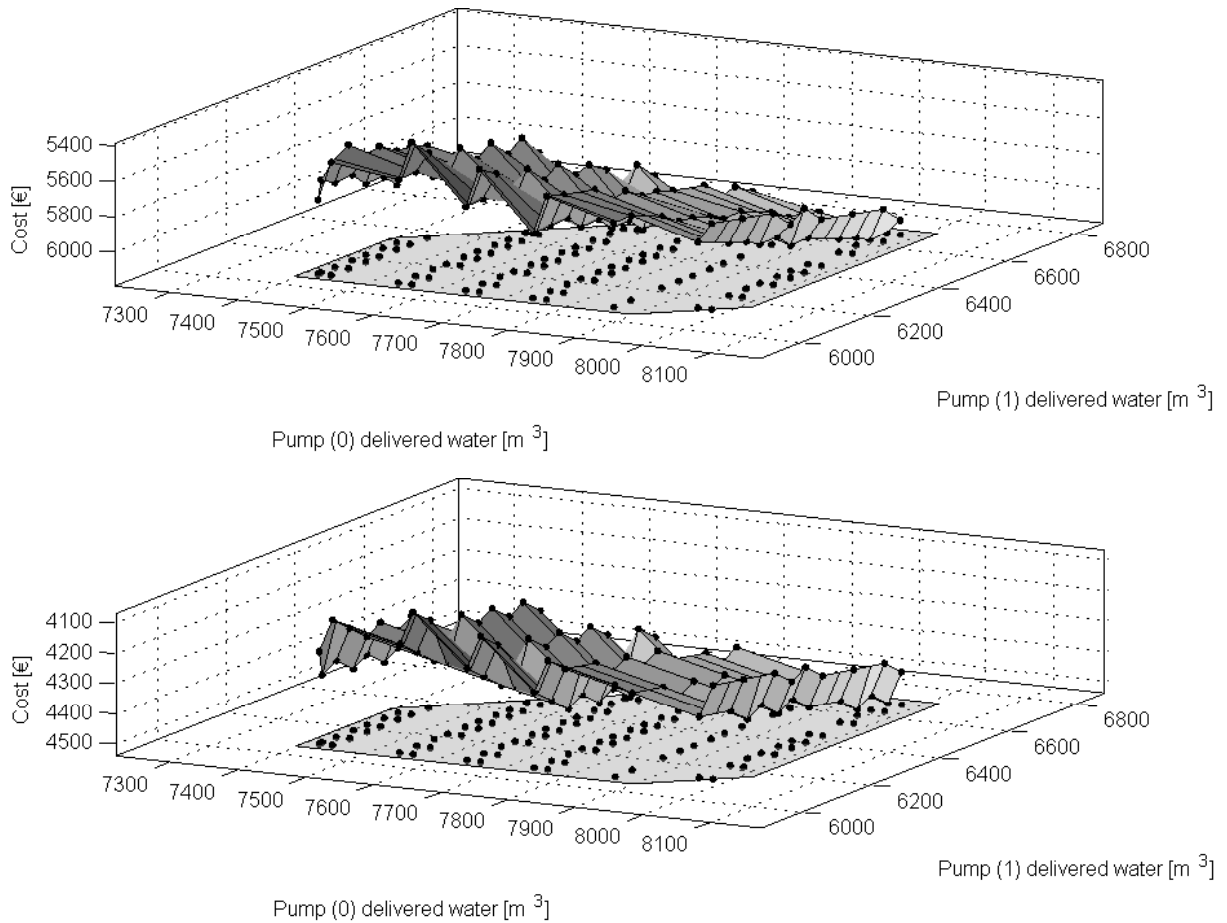


**Fig. 5.** Optimal trajectory on the pseudo state space. The grey shaded polygons express the evolution of reservoir constraints in time. The projection of the trajectory can be seen on the marginal planes.

**Tab. 5.** Comparison of the optimal solutions found by DP and DP-LP. The best solution was type-set in bold numbers.

$V_{0,1,..,23}^{min}$ [m <sup>3</sup> ]	$V_{0,24}^{min}$ [m <sup>3</sup> ]	Cost [€], DP (Discretization level below.)							Cost [€] DP-LP
		4	5	6	7	10	18	51	
100	1600	<b>5755</b>	<b>5755</b>	<b>5755</b>	<b>5755</b>	<b>5755</b>	<b>5755</b>	<b>5755</b>	<b>5755</b>
1000	1600	5920	<b>5810</b>	<b>5810</b>	<b>5810</b>	<b>5810</b>	<b>5810</b>	<b>5810</b>	<b>5810</b>
1600	1600	6115	6030	<b>5920</b>	<b>5920</b>	<b>5920</b>	<b>5920</b>	<b>5920</b>	<b>5920</b>
1700	1700	failed	6295	6245	6245	6245	6245	<b>6085</b>	<b>6085</b>



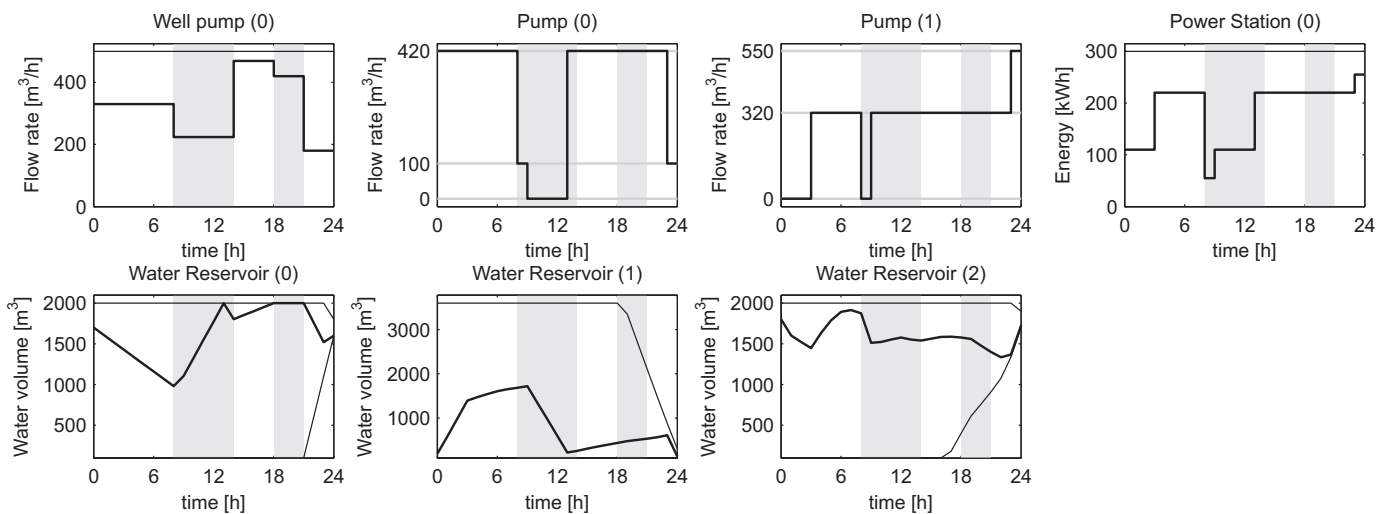


**Fig. 6.** Cost-to-go function at time period 23. Top: the original problem. Bottom: the energy tariff is uniform (1 €/kWh on the optimization horizon). Due to finite control action sets the shape of the cost-to-go function is rugged

(regardless to energy pricing) even though the objective function is a smooth, quadratic formula.

**Tab. 6.** Comparison of the optimal solutions found by the NEOS solvers and DP-LP. The best solution was typeset in bold numbers.

$V_{0,1..23}^{\min}$ [m <sup>3</sup> ]	$V_{0,24}^{\min}$ [m <sup>3</sup> ]	Cost, NEOS solvers [€]						Cost [€]
		Cbc	Glpk	Gurobi	MOSEK	scip	XpressMP	
100	1600	5830	6025	5830	5940	<b>5755</b>	5830	<b>5755</b>
1000	1600	5940	6190	<b>5810</b>	5940	6130	6100	<b>5810</b>
1600	1600	6210	6245	6085	6250	6085	5975	<b>5920</b>
1700	1700	6470	6520	<b>6085</b>	6305	<b>6085</b>	<b>6085</b>	<b>6085</b>



**Fig. 7.** Optimal schedule: 5755 €. Peak charging periods (2 €/kWh) are gray shaded while off-peak periods (1 €/kWh) are colorless. Thin lines represent constraints.

### 6.1 Modified algorithm for problem solving

To handle the continuous control variable (besides the discrete ones) the most obvious technique is the discretization of the well's flow range and the use of the obtained discrete set as control domain for the well pump. This introduces a pure combinatorial optimization problem which can be solved by the introduced dynamic programming approach (referred as DP). On the other hand discretization of the domain of the continuous variable(s) is not beneficial on real size networks due to the curse of dimensionality: the exponential growth of the size of the control domain with sub-domain cardinalities.

To overcome the difficulties an alternative solution technique is proposed observing that the well pump operation and the operation of the main distribution network can be treated separately [19]. Using this as key idea, first a candidate solution is generated for the main distribution network (just like in combinatorial case) by removing the constraint on the joint reservoir (Reservoir 0 in this particular example). This includes only the pumps of the main distribution network allowing the stored water in the joint reservoir to be outside the boundaries. At the second step a simple linear programming technique (LP solver) seeks for the possible well pump operation whether the constraints on the joint reservoir can be satisfied or not determining the feasibility of the given candidate. Feasible solutions are used to calculate the value of the cost-to-go function while infeasible ones are removed from candidate solution inventory. By repeating this process the dynamic programming recursion continues until the termination criteria is met.

Besides the simple implementation the main benefit is the unaffected search space, that is, the water supply of the main distribution network does not increase the complexity of the problem thus it is defined only by the operation of the main distribution network. The approach is able to handle numerous well inputs providing easy interpretation capabilities for real size networks as well. It assumes that the energy used for supply is not included in the objective function. This assumption however does not contradict with real life since well pumps usually represent less than 5% of the total energy consumption of a water distribution network. This small fraction of the total energy required by the water supply of the main distribution network can be simply neglected in the corresponding model.

On the other hand the achieved benefits come with a price. This makes the presented solver an approximate dynamic programming approach. 'Approximate' reflects to the fact that it does not ensure the finding of the global optimum of the underlying problem, one may get it however it is not guaranteed. We refer to this method as DP-LP.

The comparison of the obtained optimal control policies of different well handling setups (Fig. 7 and 4) highlight the fact that greater operational flexibility on well fields allows better cost reduction.

### 6.2 Results

First the detailed dynamic programming techniques (DP and DP-LP) are compared on problems defined in Section 5. Under DP the operational range of the well pump was discretized obtaining sparser and denser setups for comparison. The well pump control set configurations were as follows:  $\{0, 165, 330, 500\}[m^3/h]$ ,  $\{0, 165, 330, 415, 500\}[m^3/h]$ ,  $\{0, 110, 220, 330, 415, 500\}[m^3/h]$ ,  $\{0, 80, 160, 245, 330, 415, 500\}[m^3/h]$ ,  $\{0, 55, 110, \dots, 440; 500\}[m^3/h]$ ,  $\{0, 30, 60, \dots, 480; 500\}[m^3/h]$ ,  $\{0, 10, 20, \dots, 500\}[m^3/h]$  obtaining 4, 5, 6, 7, 10, 18 and 51 discretization levels on well pump operating range.

Table 5 summarizes the results. The conclusion is obvious: stricter constraint system requires finer discretization on the well pump operational range to achieve the global optimum. The finer the discretization is the more flexible the operation of the well becomes. In this particular example representing the well pump control set by 18 control actions seemed satisfactory. On the other hand DP-LP has obtained the attainable optimum as well on each problem instances.

In contrast Table 6 shows the achieved best solutions by NEOS solvers compared to DP-LP. It provides only an informative comparison on the performance of the approximate dynamic programming with respect to NEOS solvers on this particular problem. Finally the optimal schedule corresponding to the original problem (Tab. 6 first row) can be seen in Figure 7.

## 7 Conclusions and further work

A novel optimal control approach was introduced for operational optimization of water distribution networks having on/off type controls (discrete active hydraulic elements) and applied to a small scale sub-network of the water supply and distribution network of Sopron (Hungary). The approach resolves the problem of state space discretization and allows dynamic programming to be used as optimal control tool on combinatorial pump scheduling problems. By observing permutational symmetries in the dynamics of water distribution networks a series of pseudo states formed over control domain are used to recursively compute the Bellman equation replacing the original state space based dynamic programming procedures.

As a great benefit the finding of global optimum on combinatorial problems is always guaranteed however in its current form the usability is restricted to small and medium sized water networks due to the curse of dimensionality. Besides combinatorial optimization a combined dynamic programming-linear programming (DP-LP) technique was invoked to handle continuous decisions as well represented by the water feed of the main distribution network. By using this technique the cardinality of the control domain remains unchanged while its dimension is increasing by additional continuous control variables (represented by well pumps). Thus the complexity of the problem is determined only by the main distribution network. By partially resolving the curse of dimensionality numerous well fields can be handled. In addition these benefits are coupled with the ease of

implementation on real size networks. On the other hand DP–LP approach does not guarantee the finding of the global optimum on mixed–combinatorial problems besides that a great potential was demonstrated by the application.

All things considered the authors suggest further studies on the utilization of DP–LP on real size water distribution networks.

## References

- 1 **Bene J G, Selekk I, Hős Cs.** *Neutral Search Technique for Short-Term Pump Schedule Optimization*, *Journal of Water Resources Planning and Management* **136** (2010), no. 1, 133–137. doi:10.1061/(ASCE)0733-9496(2010)136:1(133) (5 pages).
- 2 **Ikonen E, Selekk I, Tervaskanto M.** *Short-term pump schedule optimization using MDP and neutral GA*, IFAC Conference on Control Methodologies and Technology for Energy Efficiency (CMTEE 2010) (Vilamoura, Portugal, Mar 29, 2010).
- 3 **NEOS**, *Server for Optimization*, 2011, available at <http://www.neos-server.org>.
- 4 **CBC News**, *China's supercomputer called world's fastest*, 2010, available at <http://www.cbc.ca/news/technology/story/2010/10/28/technology-china-fastest-supercomputer.html>.
- 5 **Wang J, Chang T, Chen J.** *An enhanced genetic algorithm for bi-objective pump scheduling in water supply*, *Expert Systems with Applications* **36** (2009), 10249–10258. doi:10.1016/j.eswa.2009.01.054.
- 6 **Fang H, Zhang J, Gao J.** *Optimal Operation of Multi-Storage tank multi-source system based on storage policy*, *Journal of Zhejiang University - SCIENCE A* **11** (2010), no. 8, 571–578. doi:10.1631/jzus.A0900784.
- 7 **Alandi P, Pérez P, Álvarez J, Hidalgo M, Benito J.** *Pumping Selection and Regulation for Water-Distribution Networks*, *Journal of Irrigation and Drainage Engineering* **131** (2005), no. 3, 273–281. doi:10.1061/(ASCE)0733-9437(2005)131:3(273) (9 pages).
- 8 **Kadu M, Gupta R, Bhavne P.** *Optimal Design of Water Networks Using Modified Genetic Algorithm with Reduction in Search Space*, *Journal of Water Resources Planning and Management* **134** (2008), no. 2, 147–160. doi:10.1061/(ASCE)0733-9496(2008)134:2(147) (14 pages).
- 9 **Kumar D N, Baliarsingh F.** *Folded Dynamic Programming for Optimal Operation of Multireservoir Systems*, *Water Resources Management* **17** (2003), 337–353. doi:10.1023/A:1025894500491.
- 10 **Kumar D N, Baliarsingh F, Raju K S.** *Optimal Reservoir Operation for Flood Control Using Folded Dynamic Programming*, *Water Resources Management* **24** (2010), 1045–1046. doi:10.1007/s11269-009-9485-3.
- 11 **Chandramouli V, Raman H.** *Multireservoir Modeling with Dynamic Programming and Neutral Networks*, *Journal of Water Resources Planning and Management* **127** (2001), no. 2, 89–97. doi:10.1061/(ASCE)0733-9496(2001)127:2(89) (10 pages).
- 12 **Mousavi S J, Karamouz M, Menhadj M B.** *Fuzzy-State Stochastic Dynamic Programming for Reservoir Operation*, *Journal of Water Resources Planning and Management* **130** (2004), no. 6, 460–470. doi:10.1061/(ASCE)0733-9496(2004)130:6(460) (11 pages).
- 13 **Tilmant A, Vanclooster M, Duckstein L, Persoons E.** *Comparison of Fuzzy and Nonfuzzy Optimal Reservoir Operating Policies*, *Journal of Water Resources Planning and Management* **128** (2002), no. 6, 390–398. doi:10.1061/(ASCE)0733-9496(2002)128:6(390) (9 pages).
- 14 **Cervellera C, Chen V C P, A. Wen.** *Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization*, *European Journal of Operational Research* **171** (2006), 1139–1151. doi:10.1016/j.ejor.2005.01.022.
- 15 **Sabet M H, Helweg O J.** *Cost Effective Operation of Urban Water Supply System Using Dynamic Programming*, *Water Resources Bulletin* **21** (1985), no. 1, 75–81. doi:10.1111/j.1752-1688.1985.tb05353.x.
- 16 **Sarbú I, Ostafe G.** *Energetical optimization of water distribution systems in large urban centers*, *Periodica Polytechnica Mechanical Engineering* **52** (2008), no. 2, 93–103. doi:10.3311/pp.me.2008-2.09.
- 17 **Selekk I.** *István Selekk's Website*, 2012, available at <http://www.artificialevolution.net/>.
- 18 **Castelletti A, Pianosi F, Soncini-Sessa R.** *Water reservoir control under economic, social and environmental constraints*, *Automatica* **44** (2008 June), 1595–1607, DOI 10.1016/j.automatica.2008.03.003, <http://portal.acm.org/citation.cfm?id=1377053.1377350>. doi:10.1016/j.automatica.2008.03.003.
- 19 **Pezeshek S, Helweg O J.** *Adaptive Search Optimization in Reducing Pump Operating Costs*, *Journal of Water Resources Planning and Management* **122** (1996), no. 1, 57–63. doi:10.1061/(ASCE)0733-9496(1996)122:1(57) (7 pages).
- 20 **Bellman R, Dreyfus S.** *Applied Dynamic Programming*, Princeton University Press, Princeton N. J., 1962. doi:10.1289/ehp.1002206.
- 21 **Matoušek J, Gärtner B.** *Understanding and Using Linear Programming*, Springer–Verlag, 2007. ISBN:3540306978.
- 22 **Land A H, Doig A G.** *An Automatic Method of Solving Discrete Programming Problems*, *Econometrica* **28** (1960), no. 3, 497–520.
- 23 **Labadie J W.** *Optimal Operation of Multireservoir Systems: State-of-the-Art Review*, *Journal of Water Resources Planning and Management* **130** (2004 March), no. 2, 93–111. doi:10.1061/(ASCE)0733-9496(2004)130:2(93) (19 pages).
- 24 **Halász G, Kristóf G, Kullmann L.** *Áramlás csőhálózatokban. (Flow in pipe networks)*, Műegyetemi kiadó, 2002. ISBN 9634207081.
- 25 **Chen V C P.** *Measuring the Goodness of Orthogonal Array Discretizations for Stochastic Programming and Stochastic Dynamic Programming*, *Siam Journal on Optimization* **12** (2002), no. 2, 322–344. doi:10.1137/S1052623498332403.
- 26 **Tang B.** *Orthogonal Array-Based Latin Hypercubes*, *Journal of the American Statistical Association* **88** (1993), no. 424, 1392–1397. doi:10.2307/2291282.
- 27 **Niederreiter H, Xing C.** *Low-Discrepancy Sequences and Global Function Fields with Many Rational Places*, *Finite Fields and Their Applications* **2** (1996), no. 3, 241–273. doi:10.1006/ffta.1996.0016.
- 28 ———. *Low-discrepancy sequences obtained from algebraic function fields over finite fields*, *Acta Arithmetica* **72** (1995), no. 3, 281–298. doi:10.1006/ffta.1996.0016.
- 29 **Rust J.** *Using Randomization to Break the Curse of Dimensionality*, *Econometrica* **65** (1997), no. 3, 487–516. doi:10.2307/2171751.
- 30 **Georgia V C, Chen V C P, Victoria Chen Georgia and Victoria C. P. Chen.** *Discretizing the State Space for High-Dimensional Continuous-State Stochastic Dynamic Programs*, *Mathematics of Operations Research*, 1995.
- 31 **Bene J, Hős Cs.** *Computation of Cost-optimal Pump Scheduling for Regional Waterwork using Genetic Algorithm*, Fifth Conference on Mechanical Engineering (Gépészet 2006) (Budapest, Hungary, May 25, 2006).
- 32 ———. *A Novel Constraint Handling Technique for Genetic Algorithm-based Pump Schedule Optimization*, Seventh Conference on Mechanical Engineering (Gépészet 2010) (Budapest, Hungary, May 25, 2010).
- 33 **Jensen P A, Bard J F.** *Operations research: models and methods*, Wiley, 2003. doi:10.2307/2344020.
- 34 **Koncsos L, Balogh E.** *A simulation-optimisation methodology for designing the operation of emergency reservoirs in the Hungarian Tisza basin*, *Periodica Polytechnica Civil Engineering* **54** (2010), no. 2, 101–106. doi:10.3311/pp.ci.2010-2.05.
- 35 **Darabos P.** *Software package for controlling water supply systems*, *Periodica Polytechnica Civil Engineering* **41** (1997), no. 2, 135–145.