Delft University of Technology

Delft Center for Systems and Control

Technical report 10-033

Decomposition of a fixed-profile load scheduling method for large-scale irrigation channels*

Y. Li, J. Alende, M. Cantoni, and B. De Schutter

If you want to cite this report, please use the following reference instead:
Y. Li, J. Alende, M. Cantoni, and B. De Schutter, "Decomposition of a fixed-profile load scheduling method for large-scale irrigation channels," *Proceedings of the 2010 IEEE International Conference on Control Applications*, Yokohama, Japan, pp. 2166–2171, Sept. 2010.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: `https://www.dcsc.tudelft.nl`

Decomposition of a Fixed-Profile Load Scheduling Method for Large-Scale Irrigation Channels

Yuping Li, Julien Alende, Michael Cantoni, and Bart De Schutter

Abstract— **The problem of fixed-profile load scheduling is considered for large-scale irrigation channels. Based on the analysis of the special structure of a channel under decentralised control, a predictive model is built on a pool-by-pool basis and a decomposition strategy of the scheduling problem is provided. The decomposition avoids excessive memory requirements in building the predictive model of the controlled plant and solving the formulated optimisation problem.**

Index Terms— **Fixed-profile load scheduling, $\{0, 1\}$ linear programming, large-scale systems, predictive model, constrained optimisation, hierarchical control.**

I. INTRODUCTION

In large-scale irrigation networks, water is often distributed via open water channels under the power of gravity (i.e. there is no pumping). The flow of water through the network is regulated by automated gates positioned along the channels [3], [8], [10]. The stretch of a channel between two gates is commonly called a pool. Water offtake points to farms and secondary channels are distributed along the pools.[1] As such, an important control objective is setpoint regulation of the water-levels immediately upstream of each gate, which enables flow demand at the (often gravity-powered) offtake points to be met without over-supplying. When the number of pools to be controlled is large and the gates widely dispersed, it is natural to employ a decentralised control structure. Fig. 1 shows a side view of a channel under decentralised feedback control. The flow into pool$_i$, denoted by u_i, equals to flow supplied by the upstream pool, v_{i-1}. Note u_i is actually the control action taken by controller C_i to regulate the water-level y_i to a relevant setpoint r_i,[2] in the face of disturbances associated with variations of the uncontrolled load d_i.

In practice, channel capacity is limited. This forces farmers to take water by placing orders. Moreover, the time-delay for water to travel from the upstream end to the downstream end of the pool limits the closed-loop bandwidth, which dampens the performance. Hence, the starting and ending of offtakes (d_i) induce transients (i.e. the water-level drops and rises from setpoint). Such a transient response propagates to upstream pools as regulators take corrective actions [3], [6]. Indeed, water-levels are equivalent to setpoints in steady-state. Hence, the open water channel management objectives can be expressed in terms of constraints on the water-levels in each pool: upper bounds avoid water spillage over the banks of the channel; and lower bounds ensure a minimal channel capacity to supply water. In load scheduling, a set of offtakes (requested by farmers) is organised, which ensures the water-level constraints are satisfied, in the face of transients associated with load changes. Moreover, from a farmer's perspective, a preferable solution would involve the smallest possible delay between the requested starting time and the time the load is scheduled. As a result, the scheduling can be expressed as an optimisation problem involving minimising the delay of water delivery subject to constraints.

Indeed, the load scheduling sits on the higher-level of a two-level control hierarchy. On the lower-level, controllers are designed to ensure stability, robustness, good setpoint tracking, and disturbance rejection. The following load scheduling problem is considered in this paper, in particular, for large-scale irrigation channels. Given

- a linear controlled plant whose controller rejects disturbances associated with load variation,
- linear constraints on the transient response,
- load orders from users,

determine the smallest delay between the time the load is requested to start and the time it can be scheduled, without violation of the constraints.[3] Note that preserving the profile of the requested load is a strong constraint on the scheduling task. Such a constraint corresponds to a specific production requirement, e.g. constant load over time in gravity-fed irrigation channels.

In [1], a predictive model of the controlled plant over a finite horizon is built as a function of the load to be scheduled. Then the scheduling problem is formulated as a combinatorial optimisation problem that can be rewritten as a $\{0, 1\}$ Linear Programming problem. However, when applied in load scheduling for large-scale irrigation channels, such a formulation has several limitations mainly due to computational issues with the size of the predictive model and the time to solve the constrained integer optimisation problem. In this paper, the special structure of an irrigation channel

This work was supported in part by the Australian Research Council (Linkage Grant LP0989497), the European 7th framework STREP project HD-MPC, the Delft Research Center NGI, and the BSIK project NGI.

Y. Li and B. De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, the Netherlands. Email: {yuping.li,b.deschutter}@tudelft.nl

J. Alende is a Control Engineer at Rubicon System Australia Ltd Pty. Email: julien.alende@rubicon.com.au

M. Cantoni is with The University of Melbourne, Dept. E&E Engineering, Parkville, VIC 3010, Australia. Email: cantoni@unimelb.edu.au

[1]Typically at the downstream end of pools.

[2]The water-level setpoints could be an outcome of a Supervisory Supply Management System [2]; here they are considered known.

[3]It is important to differentiate: – the (transportation) time-delay water takes to travel from the upstream to the downstream end of a pool, and – the (delivery) delay between the time an offtake is requested to start and the time it is scheduled. The first delay is a known characteristic of the system, the second one is the decision variable of the scheduling problem.

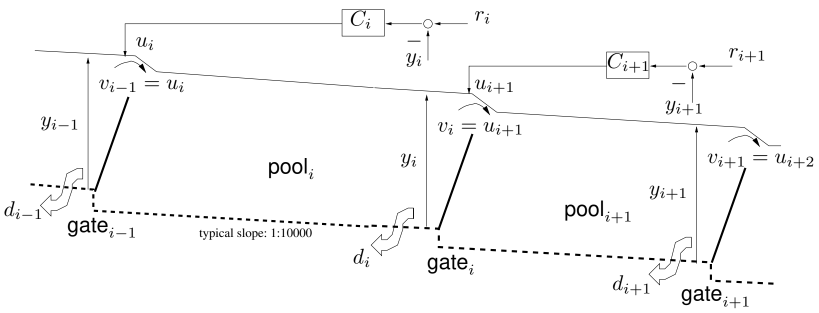

Fig. 1. Decentralised control of an open water channel

under decentralised control is studied.[4] It is shown that to overcome the associated computational issues, it is useful to build the predictive model on a pool-by-pool basis, with the interconnection between controlled pools as a constraint in the formulation of the scheduling problem. To decrease the computing time in solving the optimisation (scheduling) problem, a decomposition strategy is suggested, which decreases the number of decision variables and constraints. The resulting solution might be suboptimal compared to the nominal solution (see [1]) of the load scheduling problem.

The paper is organised as follows. Section II discusses the fixed-profile load scheduling problem for irrigation channels as formulated in [1]. By analysing the special structure of a channel under decentralised control, a decomposition of this scheduling scheme is suggested in Section III. The solution obtained via the proposed decomposition is compared to that obtained via the original formulation in [1] for an example channel. A brief summary is finally given in Section IV.

II. Fixed-profile load scheduling problem for a controlled irrigation channel

The formulation of the fixed-profile load scheduling problem is discussed in [1]. The idea is to predict the behaviour of the controlled channel (composed of N pools) over a finite horizon as a function of the delays between the requested offtakes and the scheduled ones. Throughout the prediction horizon, the water-levels are constrained. Given a cost function penalising the overall delays, the resulting formulation is a Nonlinear Integer Programming problem. Further, identifying that the number of values the delays can take in a finite horizon is finite, by applying a change of variables, the problem is formulated as a $\{0,1\}$ Linear Programming problem as follows.

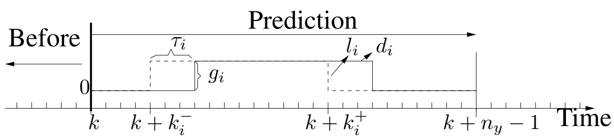

Fig. 2. A requested load and one possible schedule

Let l_i, see the dashed line in Fig. 2, denote the requested offtake from pool$_i$. Within the prediction horizon (i.e. from the time slot k to $k+n_y-1$), the profile of l_i is represented by $l_i^{[k,k+n_y-1]} = \left[\mathbf{0}^{k_i^-}, g_i \times \mathbf{1}^{k_i^+-k_i^-}, \mathbf{0}^{n_y-k_i^+}\right]^T$, where $\mathbf{0}^n$ represents a vector of n 0's and $\mathbf{1}^n$ a vector of n 1's, $k+k_i^-$ and $k+k_i^+$ denote the starting and stopping time of the requested offtake, respectively, and g_i is the magnitude of the offtake. One possible schedule of the requested offtake, d_i, is shown in the figure (see the solid line). In particular,

$$d_i^{[k,k+n_y-1]} = J^{\tau_i} l_i^{[k,k+n_y-1]},$$

where $\tau_i \in \mathbb{N}_0$ is the delivery delay between the starting time of the requested offtake l_i and the starting time of the scheduled offtake d_i;[5] and J is a $n_y \times n_y$ lower shift matrix,

$$J = \begin{bmatrix} 0 & & \cdots & 0 \\ 1 & 0 & & \vdots \\ & \ddots & \ddots & \\ 0 & & 1 & 0 \end{bmatrix}.$$

Note l_i has n_i $(= n_y - k_i^+ - T_i^{\max})$ possible schedules, where $T_i^{\max}$ is the maximal duration of the transients in pool$_1$ to pool$_i$ caused by stopping of the offtake d_i.[6] Indeed, let $M_i \in \mathbb{R}^{n_y \times n_i}$ represent all the possible delayed versions of the requested load, any schedule of l_i can be represented by

$$d_i = \left[J^0 l_i | J^1 l_i | \dots | J^{n_i-1} l_i\right] z_i =: M_i z_i, \qquad (1)$$

where z_i is a vector of size n_i, with only one element equal to 1 and the others 0, which corresponds to only one schedule being selected. For a string of N pools, the offtake-load scheduling problem is then formulated as

$$\begin{aligned} &\min_{z_i} \sum_{i=1}^{N} h_i^T z_i \\ &\text{s.t. } \hat{y}^{[k+1,k+n_y]} = \sum_{i=1}^{N} f_i(z_i) && (2) \\ &\underline{y}^{[k+1,k+n_y]} \le \hat{y}^{[k+1,k+n_y]} \le \overline{y}^{[k+1,k+n_y]}; \\ &\sum_{m=1}^{n_i} (z_i)_m = 1, \\ &z_i \in \{0,1\}^{n_i} \text{ for } i = 1,\dots,N. && (3) \end{aligned}$$

[4] In fact, the decomposition strategy proposed in this paper can also be applied to load scheduling for channels under distributed control [3], [5].

[5] In this paper, scheduling an offtake in advance of its requested time, i.e. $\tau_i < 0$, is not considered from the practical perspective.

[6] This considers the transient propagating to upstream pools under distant-downstream control as shown in Fig. 1.

In the cost function, h_i^T is a weight penalising the delays. The equality constraint (2) is a process model of the controlled channel, it predicts the water-levels of the N pools as a linear function of the decision variables z_i,[7] where $\hat{y}^{[k+1,k+n_y]} := \begin{bmatrix} \hat{y}(k+1) \\ \vdots \\ \hat{y}(k+n_y) \end{bmatrix}$ with $\hat{y}(k) = \begin{bmatrix} \hat{y}_1(k) \\ \vdots \\ \hat{y}_N(k) \end{bmatrix}$, where $\hat{y}_i(k)$ is the estimate of the water-level y_i at time k. Constraint (3) requires that the n_i elements in z_i be either 0 or 1.

Such a formulation is an MILP, for which efficient algorithms exist [4]. However, as pointed out in [1], it has several limitations mainly due to computational issues:

- The size of the predictive model is proportional to the number of pools and the length of the prediction horizon.
- The computing time when solving the constrained integer optimisation problem is polynomial in the number of decision variable and constraints.

The facts that for offtake-load scheduling the forecast horizon n_y is often large (e.g. in the simulation in [1] $n_y = 480$, to forecast a scheduling for 80 hours under a practical consideration) and that the number of pools in a channel could be above 30 make the previous scheduling strategy impractical for large-scale irrigation network. To overcome this, a decomposition strategy is suggested in the next section, which is based on the analysis of the special structure of a string of pools under decentralised control.

III. Decomposition of the load scheduling problem

For an irrigation channel under decentralised distant-downstream control, the information exchange between subsystems is one-by-one from downstream to upstream (i.e. $v_i = u_{i+1}$ as shown in Fig. 1). Such a special structure makes it feasible to schedule load requests from downstream to upstream in sequence, i.e. first schedule for d_N, then d_{N-1}, ..., at last for d_1. Hence the scheduling optimisation problem introduced in Section II can be decomposed. Moreover, in such a decomposition, the interconnections between subsystems are represented as a function of the already scheduled load. We can then build a predictive model based on each controlled pool, which demands much less memory space than that of a controlled channel [1].

A. Predictive model building for a controlled pool

A simple frequency-domain model of the water-level in pool$_i$, that is based on mass balance (see [9]) and that captures the dynamics at low frequencies, is obtained:

$$y_i(s) = \frac{c_{\text{in},i} e^{-t_{\text{d},i} s}}{s} u_i(s) - \frac{c_{\text{out},i}}{s} d_i(s) - \frac{c_{\text{out},i}}{s} v_i(s),$$

where $c_{\text{in},i}$ and $c_{\text{out},i}$ are discharge coefficients, functions of the pool surface area and the gate width; and $t_{\text{d},i}$ is the internal time-delay that the water takes to travel from the upstream end to the downstream end of a pool. Essentially, the decentralised controller C_i is a PI compensator with low-pass filter [3], [10]:

$$u_i(s) = \frac{\kappa_i}{\phi_i} \frac{(1+s\phi_i)}{s(1+s\rho_i)} (r_i(s) - y_i(s)),$$

in which the integrator is involved for zero steady-state water-level error in rejection to step load disturbance d_i, the low-pass filter ensures no excitement of (unmodelled) dominant wave dynamics, while the phase-lead term helps for closed-loop stability. Then a continuous state-space realisation of the controlled pool is obtained by using a first-order Padé approximation to represent the transportation time-delay $t_{\text{d},i}$:

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_{r_i} r_i(t) + B_{d_i} d_i(t) + A_{p_i} v_i(t) \\ y_i(t) &= C_i x_i(t) \end{aligned}$$

where $A_i = \begin{bmatrix} 0 & c_{\text{in},i} & -c_{\text{in},i} & 0 \\ 0 & \frac{-2}{t_{\text{d},i}} & \frac{4}{t_{\text{d},i}} & 0 \\ \frac{-\kappa_i}{\rho_i} & 0 & 0 & 1 \\ \frac{-\kappa_i(\rho_i - \phi_i)}{\phi_i \rho_i^2} & 0 & 0 & \frac{-1}{\rho_i} \end{bmatrix}$, $A_{p_i} = \begin{bmatrix} -c_{\text{out},i} \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $B_{d_i} = \begin{bmatrix} -c_{\text{out},i} \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $B_{r_i} = \begin{bmatrix} 0 \\ 0 \\ \frac{\kappa_i}{\rho_i} \\ \frac{\kappa_i(\rho_i - \phi_i)}{\phi_i \rho_i^2} \end{bmatrix}$, $C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$. Note that the interconnection between neighbouring (controlled) pools can be expressed as $v_i = u_{i+1}$. Indeed, u_i can be expressed by the following state-space form of controller

$$\begin{aligned} \dot{x}_i^K(t) &= A_i^K x_i^K(t) + B_{e_i}(r_i(t) - y_i(t)) \\ u_i(t) &= C_i^K x_i^K(t) \end{aligned}$$

where $A_i^K = \begin{bmatrix} 0 & \frac{\kappa_i(\rho_i - \phi_i)}{\phi_i \rho_i^2} \\ 0 & \frac{-1}{\rho_i} \end{bmatrix}$, $B_{e_i} = \begin{bmatrix} \frac{\kappa_i}{\rho_i} \\ 1 \end{bmatrix}$ and $C_i^K = \begin{bmatrix} 1 & 0 \end{bmatrix}$. To build the predictive model, a discrete-time state-space model (4-5) is employed. This can be obtained by direct converting the continuous model through a zero-order hold. The sampling interval T_s should be of duration small enough to capture the whole relevant dynamics of the system.

$$\begin{bmatrix} x_i(k+1) \\ y_i(k) \end{bmatrix} = \begin{bmatrix} \bar{A}_i & \bar{B}_{r_i} & \bar{B}_{d_i} & \bar{A}_{p_i} \\ \bar{C}_i & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i(k) \\ r_i(k) \\ d_i(k) \\ v_i(k) \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} x_i^K(k+1) \\ u_i(k) \end{bmatrix} = \begin{bmatrix} \bar{A}_i^K & \bar{B}_{e_i} \\ \bar{C}_i^K & 0 \end{bmatrix} \begin{bmatrix} x_i^K(k) \\ r_i(k) - y_i(k) \end{bmatrix} \quad (5)$$

The predictions of the response of a controlled pool over a finite horizon of n_y slots (of duration T_s) (i.e. from the instant $k+1$ to the instant $k+n_y$) can be computed as follows, by writing the dynamic equation of the discrete model (4-5) recursively as discussed in [7]:

$$\begin{aligned} y_i^{[k+1,k+n_y]} &= \Gamma_i x_i(k) + \Omega_i r_i^{[k,k+n_y-1]} + \Psi_i \tilde{d}_i^{[k,k+n_y-1]} \\ &+ \Psi_i d_i^{[k,k+n_y-1]} + \Upsilon_i v_i^{[k,k+n_y-1]} \text{ for } i = 1, \ldots, N \quad (6) \\ v_i^{[k+1,k+n_y]} &= \Gamma_{i+1}^K x_{i+1}^K(k) + \Pi_{i+1} r_{i+1}^{[k,k+n_y-1]} \\ &- \Pi_{i+1} y_{i+1}^{[k,k+n_y-1]} \text{ for } i = 1, \ldots, N-1 \quad (7) \end{aligned}$$

[7] f_i is a linear matrix function.

where $\Gamma_i = \begin{bmatrix} \bar{C}_i\bar{A}_i \\ \bar{C}_i\bar{A}_i^2 \\ \vdots \\ \bar{C}_i\bar{A}_i^{ny} \end{bmatrix}$, $\Gamma_{i+1}^K = \begin{bmatrix} \bar{C}_{i+1}^K\bar{A}_{i+1}^K \\ \bar{C}_{i+1}^K(\bar{A}_{i+1}^K)^2 \\ \vdots \\ \bar{C}_{i+1}^K(\bar{A}_{i+1}^K)^{ny} \end{bmatrix}$,

$$\Omega_i = \begin{bmatrix} \bar{C}_i\bar{B}_{r_i} & & & \\ \bar{C}_i\bar{A}_i\bar{B}_{r_i} & \bar{C}_i\bar{B}_{r_i} & & \\ \vdots & \vdots & \ddots & \\ \bar{C}_i\bar{A}_i^{(n_y-1)}\bar{B}_{r_i} & \bar{C}_i\bar{A}_i^{(n_y-2)}\bar{B}_{r_i} & \cdots & \bar{C}_i\bar{B}_{r_i} \end{bmatrix},$$

$$\Psi_i = \begin{bmatrix} \bar{C}_i\bar{B}_{d_i} & & & \\ \bar{C}_i\bar{A}_i\bar{B}_{d_i} & \bar{C}_i\bar{B}_{d_i} & & \\ \vdots & \vdots & \ddots & \\ \bar{C}_i\bar{A}_i^{(n_y-1)}\bar{B}_{d_i} & \bar{C}_i\bar{A}_i^{(n_y-2)}\bar{B}_{d_i} & \cdots & \bar{C}_i\bar{B}_{d_i} \end{bmatrix},$$

$$\Upsilon_i = \begin{bmatrix} \bar{C}_i\bar{A}_{p_i} & & & \\ \bar{C}_i\bar{A}_i\bar{A}_{p_i} & \bar{C}_i\bar{A}_{p_i} & & \\ \vdots & \vdots & \ddots & \\ \bar{C}_i\bar{A}_i^{(n_y-1)}\bar{A}_{p_i} & \bar{C}_i\bar{A}_i^{(n_y-2)}\bar{A}_{p_i} & \cdots & \bar{C}_i\bar{A}_{p_i} \end{bmatrix}, \quad \Pi_{i+1} =$$

$$\begin{bmatrix} \bar{C}_{i+1}^K\bar{B}_{e_{i+1}} & & & \\ \bar{C}_{i+1}^K\bar{A}_{i+1}^K\bar{B}_{e_{i+1}} & \bar{C}_{i+1}^K\bar{B}_{e_{i+1}} & & \\ \vdots & \vdots & \ddots & \\ \bar{C}_{i+1}^K(\bar{A}_{i+1}^K)^{(n_y-1)}\bar{B}_{e_{i+1}} & \bar{C}_{i+1}^K(\bar{A}_{i+1}^K)^{(n_y-2)}\bar{B}_{e_{i+1}} & \cdots & \bar{C}_{i+1}^K\bar{B}_{e_{i+1}} \end{bmatrix}.$$

Note that in (6) the applied load consists of $\tilde{d}_i$, representing the already scheduled load, and d_i, representing the load to be scheduled. It is assumed that a scheduled offtake will be executed as it is planned, hence the already scheduled loads influence the scheduling result of the newly requested loads.

In fact, for large-scale irrigation networks, the memory space gained by building the model on a pool-by-pool basis is substantial compared with that on a channel basis. In particular, the transfer function matrix from d_i to y_i is represented by a block Ψ_i (see (6)); while the impact on y_i by interaction between controlled pools, i.e. v_i, is represented by a block Υ_i, which has a similar structure as Ψ_i. In total, to represent $d := [d_1, \ldots, d_N]^T$ to $y := [y_1, \ldots, y_N]^T$ for N controlled pools, $2N-1$ such blocks are requested. In contrast, in the predictive model constructed as in [1] for a channel under decentralised control, such a relationship is represented by a block-triangular matrix Ψ, which consists of $\frac{N(N+1)}{2}$ such (nonzero) blocks, each with the same size as Ψ_i.

To reduce the computational complexity, the subsequent decomposition scheme further exploits the nature of transient propagation in the upstream direction under decentralised distant-downstream control, to arrive at a sequence of smaller (in the number of decision variables and constraints) problems that produce a potentially suboptimal solution in the sense that priority is effectively given to downstream load.

B. Decomposition of the fixed-profile load scheduling problem

As previously mentioned, in distant-downstream control, when offtakes start or stop in one pool, a transient deviation of the water-level from the setpoint is expected. The control action to compensate such an influence causes the undesirable transient propagating to upstream pools. Hence, scheduling of offtakes in a pool will affect scheduling results of requested loads in all the pools upstream of the pool. As such, when considering scheduling load pool by pool, it is natural to have a scheduling sequence from d_N to d_1. So when scheduling d_i, assume $d_1, \cdots, d_{i-1}$ equal to 0. The idea is to represent the interconnection between two pools as a function of the already scheduled offtakes in the downstream pools. For example, from (7), the interconnection v_{i-1} is expressed as a function of y_i, which is a function of d_i (see (6)),[8] then from (6), y_{i-1} can be written as a function of d_i by direct substitution of variables. In this way, $y_1, \cdots, y_i$ are written as a function of d_i. Then solve for feasible d_i such that the upper-bounds and lower-bounds of y_1 to y_i are satisfied. Note that such a decomposition gives priority to offtake requests in the downstream pools.

In summary, based on the process model (6-7), let $\tilde{y}_i$ and $\tilde{v}_{i-1}$ represent the predictions of the water-level transients and the control actions in response to the already scheduled offtakes in pool$_i$, respectively:

$$\begin{aligned} \tilde{y}_i^{[k+1,k+n_y]} &:= \Gamma_i x_i(k) + \Omega_i r_i^{[k,k+n_y-1]} \\ &\quad + \Psi_i \tilde{d}_i^{[k,k+n_y-1]}, \quad (8) \\ \tilde{v}_{i-1}^{[k+1,k+n_y]} &:= \Gamma_i^K x_i^K(k) + \Pi_i\left(r_i - \tilde{y}_i\right)^{[k,k+n_y-1]} \quad (9) \end{aligned}$$

we can have the decomposition of the offtake load scheduling as the following algorithm.

1) $i \leftarrow N$; $v_N^{[k,k+n_y-1]} \leftarrow \mathbf{0}^{n_y}$.[9]
2) Set $d_1, \ldots, d_{i-1}$ equal to $\mathbf{0}^{n_y}$; represent $y_1^{[k+1,k+n_y]}$ to $y_i^{[k+1,k+n_y]}$ as a function of $d_i^{[k,k+n_y-1]}$, as given by the following steps a) to d).
   a) From (6) and (8),
   $$\begin{aligned} y_i^{[k+1,k+n_y]} &\leftarrow \tilde{y}_i^{[k+1,k+n_y]} + \Upsilon_i v_i^{[k,k+n_y-1]} \\ &\quad + \Psi_i d_i^{[k,k+n_y-1]}; \quad (10) \end{aligned}$$
   b) Set $j \leftarrow i-1$;
   c) From (6) and (8) and the assumption that for $j = 1, \ldots, i-1$, $d_j = \mathbf{0}^{n_y}$,
   $$y_j^{[k+1,k+n_y]} \leftarrow \tilde{y}_j^{[k+1,k+n_y]} + \Upsilon_j v_j^{[k,k+n_y-1]},$$
   where $v_j^{[k,k+n_y-1]} \leftarrow \begin{bmatrix} u_{j+1}(k) \\ v_j^{[k+1,k+n_y-1]} \end{bmatrix}$ with
   $$\begin{aligned} v_j^{[k+1,k+n_y]} &\leftarrow \tilde{v}_j^{[k+1,k+n_y]} - \Pi_{j+1} y_{j+1}^{[k,k+n_y-1]} \\ &\quad + \Pi_{j+1}\tilde{y}_{j+1}^{[k,k+n_y-1]}. \end{aligned}$$
   Note that $u_{j+1}(k)$ can be calculated directly from the initial state of the system, see (5).
   d) $j \leftarrow j-1$; if $j = 0$, end; else go to c).
3) Represent y_1 to y_i as functions of z_i, by replacing d_i in (10) with $M_i z_i$ as in (1).

[8] v_i is known: it is a function of the already scheduled $\tilde{d}_{i+1}$.

[9] Under distant-downstream control, setting a boundary condition of $v_N = 0$ is indeed possible [3].

4) Find optimal z_i satisfying

$$\begin{aligned}&\min_{z_i} h_i^T z_i\\ &\text{s.t. for all } h=1,\ldots,i\\ &\underline{y}_h^{[k+1,k+n_y]} \le y_h^{[k+1,k+n_y]} \le \overline{y}_h^{[k+1,k+n_y]}\\ &\sum_{m=1}^{n_i} (z_i)_m = 1\\ &z_i \in \{0,1\}^{n_i}.\end{aligned}$$

If no feasible solution found, $d_i^{[k,k+n_y-1]} \leftarrow \mathbf{0}^{n_y}$.

5)

$$v_{i-1}^{[k,k+n_y-1]} \leftarrow \begin{bmatrix} u_i(k) \\ v_{i-1}^{[k+1,k+n_y-1]} \end{bmatrix};\ \text{in which}$$

$v_{i-1}^{[k+1,k+n_y]} \leftarrow \tilde{v}_{i-1}^{[k+1,k+n_y]} - \Pi_i (y_i - \tilde{y}_i)^{[k,k+n_y-1]}$ where $(y_i - \tilde{y}_i)^{[k,k+n_y-1]} \leftarrow \begin{bmatrix} 0 \\ (y_i-\tilde{y}_i)^{[k+1,k+n_y-1]} \end{bmatrix}$,[10] with

$$(y_i - \tilde{y}_i)^{[k+1,k+n_y]} \leftarrow \Psi_i d_i^{[k,k+n_y-1]} + \Upsilon_i v_i^{[k,k+n_y-1]}$$

6) $i \leftarrow i-1$; if $i=0$, end; else, go to 2).

Note that at time k, the plant output $y_i(k)$ and the controller output $u_i(k)$ are represented in the initial states $x_i(k)$ and $x_i^K(k)$, which could be an estimation of the states, i.e. $\hat{x}_i(k)$ and $\hat{x}_i^K(k)$.

On the one hand, the computational complexity of the above procedure is light compared with that of the scheduling scheme proposed in [1], without considering the special structure of the controlled plant. For example, to schedule N water demands from N pools (i.e. one demand per pool), each requested load, l_i, has n_i possible schedules. Hence the number of decision variables for each demand is n_i. For the scheduling scheme in [1], the number of decision variables is $\sum_{i=1}^N n_i$; the number of constraints is $\sum_{i=1}^N n_i + N \times 2n_y + N$ (see the optimisation formulation introduced in Section II); and there are $\prod_{i=1}^N n_i$ combinations of possible schedules. While for the above scheduling strategy, the one with the heaviest computation load is to schedule for l_N, i.e. the load request from the last pool. In such a case the number of decision variables is n_N; the number of constraints is $n_N + N \times 2n_y + 1$; the number of the combinations of all possible schedules is n_N.

On the other hand, the combinational number of the total possible schedules from the above scheduling method is $\sum_{i=1}^N n_i$, which is small compared to that of the scheduling scheme in [1] (i.e. $\prod_{i=1}^N n_i$). In fact, the above decomposition of the scheduling problem makes the solution suboptimal compared to the optimal solution, if any existing, obtained by the scheduling scheme in [1].

C. Simulation results

The scheduling procedure introduced in Section III-B is applied in the following case studies.

First, schedule offtake requests from the last two pools (i.e. Campbells and Schifferlies) of the East Goulburn Main (EGM) channel, Victoria, Australia. The parameters of controlled pools are given in Table I. In this case, 3 requested offtakes in each pool is considered. The notation $d_{i,j}$ represents the $j-$th offtake happening in pool$_i$. Since all load-disturbances happening in the same pool are identically modelled, it is relevant to adapt the notation used in the optimisation formulation in step 4) and step 5) as follow: the overall load of a pool is noted $d_i = \sum_{j=1}^3 d_{i,j}(t)$; $M_{i,j}$ and $z_{i,j}$ are respectively the selection of eligible solution and the $\{0,1\}$ decision variable associated to the requested offtake $l_{i,j}$. The cost function is chosen as $\min_{z_{i,j}} \sum_{j=1}^3 \left[\, 1\ 2\ \cdots\ n_{i,j} \,\right] z_{i,j}$, which minimises the overall delivery delay in a pool.

Pool	$c_{\text{in},i}$	$c_{\text{out},i}$	τ_i
Campbells	0.055	0.036	5 min
Schifferlies	0.017	0.026	6 min
Controller	κ_i	ϕ_i	ρ_i
1	0.74	71.83	8.52
2	1.19	141.27	16.75

TABLE I
PARAMETERS OF (CONTROLLED) POOLS

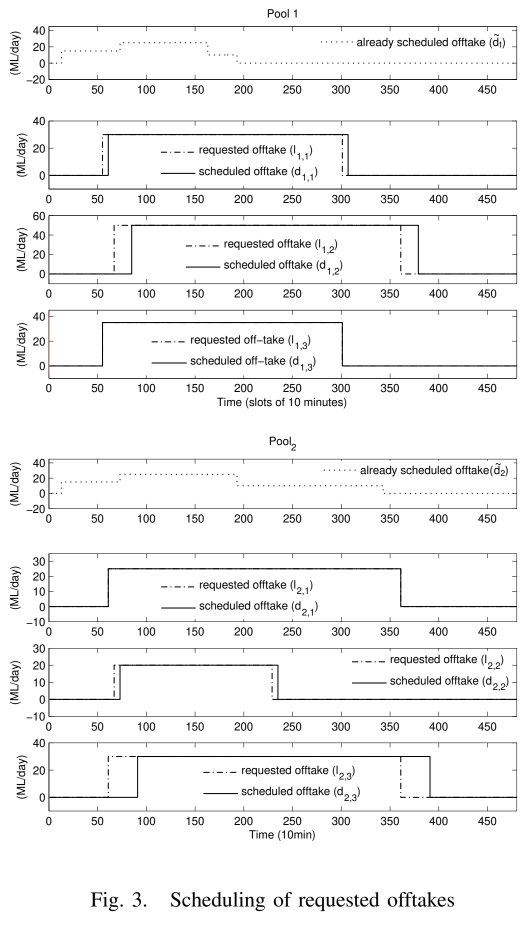

Fig. 3. Scheduling of requested offtakes

The sampling time is set to 10 minutes. The prediction horizon $n_y = 480$ (of 10 minutes), hence a forecast of 80 hours. The possible scheduled delays for each offtake is restricted to multiples of one hour (i.e. 6 ($\times$10min)). In

[10] From (4), $y_i(k) = \tilde{y}_i(k) = \bar{C}_i x_i(k)$ and hence $(y_i - \tilde{y}_i)(k) = 0$.

the simulation, the system is at steady-state at the beginning of the horizon. Given the already scheduled offtakes ($\tilde{d}_1^{[k,k+n_y]}$ and $\tilde{d}_2^{[k,k+n_y]}$ shown as in the top window of Fig. 3 respectively), schedule three requested offtakes per pool (represented by the dash-dotted line in the figure) such that the constraints on the water-level of each pool are satisfied. The requested flow is in the range of $20-30$ Ml/day, which is reasonable considering the pool characteristics. The lower bound and the upper bound on the water-levels are fixed, at 9.4 m and 9.7 m for pool$_1$ and 9.5 m and 9.7 m for pool$_2$, throughout the horizon. The setpoint r_i changes from 9.5 m to 9.6 m at 3600 min for pool$_1$ and from 9.56 m to 9.62 m at 1200 min for pool$_2$.[11]

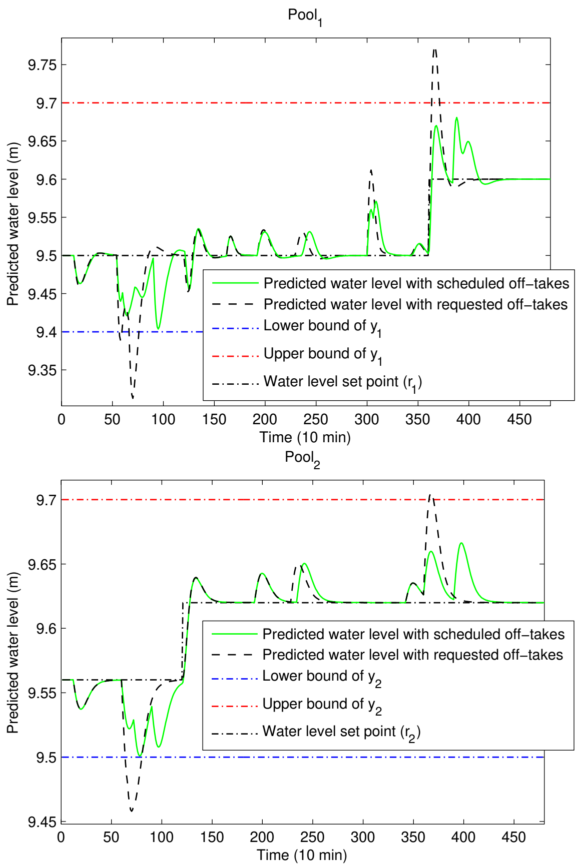

Fig. 4. Forecasting of water-levels

Forecasting of the influence of the *requested* offtakes on the system dynamic is represented by the dashed line in Fig. 4. The upper bound and lower bound (constraints) on the water-levels ($\underline{y}_i$ and $\bar{y}_i$) are violated at some time instants (around 750 min and around 3750 min) in the prediction horizon. In contrast, under the scheduled offtakes, the dynamics of the system is within the water-level constraints (see solid line in Fig. 4). The scheduling results are shown by the solid line in Fig. 3. The total time-delay (11 hours) for the 6 offtakes requests is the same as the scheduling result in [1].

[11] For comparison, the simulation scenario is set the same as that in [1].

We then check the case of scheduling for more pools. In [1], building the predictive model for a channel of 10 (controlled) pools results in a memory problem (on a Pentium4 CPU 2.8GHz, with 512 MB of RAM). By contrast, a scheduling for 10 pools by the decomposition strategy does not have such a problem. In this case, the time to build the model with a prediction of 80 hours is 20 seconds (actually 2 seconds for each controlled pool). In the simulation, the total computing time when solving the constrained mixed-integer optimisation problem (scheduling for 3 offtakes per pool) is 27 seconds, with the most complex sub-problem (scheduling for offtakes in pool$_{10}$: 9216 constraints, 105 decision variables) costing 7 seconds for solution.

IV. Conclusions

The problem of load scheduling for large-scale irrigation channels is considered. Based on the analysis of the special structure of open water channels under decentralised control, a decomposition of the scheduling problem is discussed. The solution could be suboptimal compared to an optimal solution, if it exists, to the scheduling problem initially formulated in [1], without considering the structure of the irrigation system. However, such a decomposition scheme avoids computational issues, including memory requirements and computing time, which is significant for large-scale system.

Future research can extend to integrating the scheduling scheme in a receding horizon perspective. Such an extension may decrease the conservativeness of the sub-optimal solution by implementing the decomposing scheme of the load scheduling. Indeed, by receding the prediction horizon, new offtake requests are involved in the scheduling procedure, which introduces additional scheduling combinations.

References

[1] J. Alende, Y. Li, and M. Cantoni, “A {0,1} linear program for fixed-profile load scheduling and demand management in automated irrigation channels,” in *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009, pp. 597–602.

[2] M. Cantoni, P. Dower, and E. Weyer, “Managing Australia’s water resources: Automated demand scheduling and supply control for large scale irrigation networks,” University of Melbourne, Technical Report, 2008.

[3] M. Cantoni, E. Weyer, Y. Li, S. K. Ooi, I. Mareels, and M. Ryan, “Control of large-scale irrigation networks,” *Special Issue of the Proceedings of the IEEE on the Emerging Technology of Networked Control Systems*, vol. 95, no. 1, pp. 75–91, 2007, invited.

[4] C. A. Floudas, *Nonlinear and Mixed-Integer Optimization*. Oxford University Press, 1998.

[5] Y. Li and M. Cantoni, “Distributed controller design for open water channels,” in *Proceedings of the 17th IFAC World Congress*, Seoul, South Korea, July 2008, pp. 10 033–10 038.

[6] Y. Li, M. Cantoni, and E. Weyer, “On water-level error propagation in controlled irrigation channels,” in *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference 2005*, Seville, Spain, December 2005, pp. 2101–2106.

[7] J. M. Maciejowski, *Predictive Control with Constraints*. England: Prentice Hall, 2002.

[8] I. Mareels, E. Weyer, S. K. Ooi, M. Cantoni, Y. Li, and G. Nair, “Systems engineering for irrigation systems: Successes and challenges,” *Annual Reviews in Control*, vol. 29, no. 2, pp. 191–204, August 2005.

[9] E. Weyer, “System identification of an open water channel,” *Control Engineering Practice*, vol. 9, no. 12, pp. 1289–1299, 2001.

[10] ——, “Control of irrigation channels,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, 2008.