52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

Computation Time Analysis of a Distributed Optimization Algorithm Applied to Automated Irrigation Networks

Alireza Farhadi, Michael Cantoni, and Peter M. Dower

***Abstract*— This paper considers the computation time of two algorithms for solving a structured constrained linear optimal control problem with finite horizon quadratic cost within the context of automated irrigation networks. The first is a standard centralized algorithm based on the interior point method that does not exploit problem structure. The second is distributed and based on a consensus algorithm, not specifically tailored to account for system structure, but devised rather to facilitate the management of conflicting computational and communication overheads. It is shown that there is a significant advantage in terms of computation time in using the second algorithm in large-scale networks. Specifically, for a fixed horizon length the computation time of the centralized algorithm grows as $\mathcal{O}(n^4)$ with the number n of sub-systems. By contrast, it is observed via a combination of analysis and experiment that the computation time of the distributed algorithm grows as $\mathcal{O}(n)$ with the number n of sub-systems.**

I. INTRODUCTION

A. MOTIVATION and BACKGROUND

Some large-scale systems and networks, such as automated irrigation networks, have a cascade topology. The dynamical behavior of such cascade networks can be modeled by n distributed interacting linear time invariant sub-systems of the following form:

$$\begin{aligned} S_i : x_i[k+1] &= A_i x_i[k] + B_i u_i[k] + F_i d_i[k] + v_i[k], \\ y_i[k] &= C_i x_i[k],\ z_i[k] = D_i x_i[k], \end{aligned} \tag{1}$$

for $i = 1, 2, ..., n$ and $k \in \{0, 1, 2, ..., N-1\}$, where $N \in \mathbb{N}_+$ is the horizon length, $v_i[k] = M_i x_{i+1}[k]$ represents the cascade interconnection, $x_i \in \mathbb{R}^{n_i}$ is the state variable of dimension $n_i \in \mathbb{N}_+$, $u_i \in \mathbb{R}$ is the control input, $y_i \in \mathbb{R}$ and $z_i \in \mathbb{R}$ are variables to be controlled, and $d_i \in \mathbb{R}$ is a known disturbance for the i-th sub-system. For the system (1) we are interested in solving the following linear-quadratic (LQ) constrained optimal control problem:

$$\begin{aligned} &\min_{\mathbf{u}=(u_1,...,u_n)} J(\mathbf{x}[0], \mathbf{d}, \mathbf{r}, \mathbf{u}) \\ &\text{subject to (1) and} \\ &\left\{ \begin{array}{c} y_i[k], u_i[k] \in [L_i, H_i] \\ z_i[k] \in [E_i, Z_i] \end{array} \right\} \forall i \in [1, n],\ k \in [0, N-1], \end{aligned} \tag{2}$$

where

$$\begin{aligned} &J(\mathbf{x}[0], \mathbf{d}, \mathbf{r}, \mathbf{u}) \\ &\doteq \sum_{i=1}^{n} \sum_{k=0}^{N-1} ||y_i[k] - r_i||_Q^2 + ||u_i[k] - u_i[k-1]||_R^2 \\ &\qquad + ||z_i[k]||_P^2. \end{aligned} \tag{3}$$

Here $||.||$ denotes the Euclidean norm (i.e., $||z||_P^2 \doteq z'Pz$), $\mathbf{x}[\mathbf{0}] \doteq (\, x_1'[0] \ \ . \ \ . \ \ . \ \ x_n'[0] \,)'$ is the vector of initial states, $\mathbf{d}[k] = (\, d_1[k] \ \ . \ \ . \ \ . \ \ d_n[k] \,)'$ is a vector of known disturbances, $\mathbf{r} = (\, r_1 \ \ \ldots \ \ r_n \,)'$ is the vector of desired values for y_i, and $Q, P \geq 0$, $R > 0$ are weighting matrices.

By expanding the state vector $\mathbf{x}[k] \doteq (\, x_1'[k] \ \ . \ \ . \ \ . \ \ x_n'[k] \,)'$ at time step k in terms of the initial states, disturbances and controls vectors and substituting it in the quadratic cost functional (3), the equality constraint in the optimization problem (2) vanishes and it is observed that the Quadratic Programming (QP) problem (2) involves $n_d = nN$ decision variables and $n_c = 6nN$ inequality constraints. Algorithms, such as the generic interior point method [1], involving one decision maker that is responsible for computing the controls u_i without accounting for system structure are referred to here as centralized optimization algorithms. In large-scale systems (i.e. when nN is large), the total number of constraints and decision variables can be very large. In many cases this means the computation time of the centralized optimization algorithms is not practical. Towards overcoming this computational scalability problem, in [2], [3], [4], [5] distributed optimization algorithms are proposed that exploit the computational power very often available at distributed sub-systems. These distributed algorithms can be used to approximate the solution of the structured optimal control problem (2). While the approaches do not directly account for network structure, the distribution of the computational load between sub-systems can give rise to significant reduction in computation time as studied within the context of a cascade network structure.

B. PAPER CONTRIBUTIONS

This paper compares the computation time of the centralized optimization algorithm, with the computation time of the distributed optimization algorithm of [2], for the QP problem (2) subject to inequality constraints. It is shown for a fixed horizon length that the computation time of the centralized algorithm grows as $\mathcal{O}(n^4)$ with the number n of sub-systems. By contrast, it is observed via a combination of analysis and experiment that the computation time of

*This work is supported by an Australian Research Council (ARC) Linkage Project LP0989497.

The authors thank Rubicon Water Australia for information on the East Goulburn secondary irrigation channel No. 12 used in this study.

A. Farhadi is an Assistant Professor in the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. Email: afarhadi@sharif.edu. M. Cantoni and P. M. Dower are with the Department of Electrical and Electronic Engineering, University of Melbourne, Victoria, Australia. E-mail: {cantoni,pdower}@unimelb.edu.au

the distributed algorithm grows as $\mathcal{O}(n)$ with the number n of sub-systems. On the other hand, it is shown for a fixed number of sub-systems that the computation time of the centralized algorithm grows as $\mathcal{O}(N^5)$ with the horizon length N. For a fixed number of sub-systems it is observed that the computation time of the distributed algorithm also grows as $\mathcal{O}(N^5)$ with the horizon length N. However, on the basis of experiments for a particular irrigation network of interest to us, it is observed that the growth of the distributed algorithm computation time in N is bounded above by that of the centralized algorithm. In summary, either way there is a significant advantage in terms of the computation time in using the distributed optimization algorithm of [2] in large-scale irrigation networks by exploiting the computational power very often available at distributed sub-systems and the distribution of the computational load between them.

C. PAPER ORGANIZATION

The paper is organized as follows: Section II recalls the distributed optimization algorithm of [2]. This is followed by computation time analysis for networks with cascade structure in Section III. Simulation results are presented in Section IV and the paper is concluded in Section V.

II. CONSENSUS BASED DISTRIBUTED OPTIMIZATION ALGORITHM

The consensus based distributed optimization algorithm of [2] is concerned with n interacting linear sub-systems: S_1, S_2, ... , S_n each equipped with a decision maker with limited available computational power. For this system, [2] is concerned with the following optimization problem via distributed decision makers.

$$\min_{\mathbf{u}} \Big\{ J(\mathbf{g}, \mathbf{u}),\ \mathbf{u} = (u_i' ... u_n')', u_i \in \mathcal{U}_i,\ i \in \{1, ..., n\} \Big\},$$

where $\mathbf{g}$ is a collection of known vectors, e.g., $\mathbf{g}$ may include vectors $\mathbf{x}[0]$, $\mathbf{d}$, $\mathbf{r}$, u_i is the decision variable of sub-system S_i, $\mathcal{U}_i$ is the corresponding closed convex constraint set and J is a finite horizon quadratic cost function of decision variables.

Remark 2.1: For a given vector $\mathbf{x}[0]$ of initial states, vector $\mathbf{d}[k]$ of disturbances and vector $\mathbf{r}$ of references, the cost functional J in (2) is a quadratic function of the control inputs u_i, $i \in \{1, 2, ..., n\}$. Moreover, since $[L_i, H_i], [E_i, Z_i] \subset \mathbb{R}$ are closed convex sets, and linear transformations preserve closedness and convexity [1], the control inputs (i.e. decision variables) in the QP problem (2), belong to closed convex constraint sets. Hence, the consensus based distributed optimization algorithm of [2] converges when applied to the QP (2), see [2].

In [2], decision variables u_i, $i \in \{1, ..., n\}$, are generated using an algorithm that employs the following two steps. Note that for the simplicity of presentation in the following we drop the dependency of the cost function to $\mathbf{g}$ and we present it as $J(\mathbf{u}) = J(u_1, ..., u_n)$.

- *Initialization:* In this step first each decision maker S_j chooses an arbitrary admissible value $u_j^0 \in \mathcal{U}_j$ for its decision variable. Then, the information exchange between decision makers at inner iterate $t = 0$ makes it possible for each sub-system $\mathcal{S}_i$ to initialize its local decision variables as u_i^0, $\forall i \in \{1, ..., n\}$, where $u_i^0 \in \mathcal{U}_i$ are chosen arbitrarily at $t = 0$.
- *Inner Iterate:* Then, sub-system $\mathcal{S}_i$ performs the following tasks iteratively:
  Sub-system S_i first updates its decision variable via

$$u_i^{t+1} = \pi_i u_i^* + (1 - \pi_i) u_i^t, \tag{4}$$

  where π_i is chosen subject to $\pi_i > 0$, $\sum_{i=1}^n \pi_i = 1$, and $u_i^* \doteq \text{argmin}_{u_i \in \mathcal{U}_i} J(u_1^t, ..., u_i, ..., u_n^t)$. Then, it shares its updated decision variable, u_i^{t+1} with all other sub-systems.

In [2] the authors proved feasibility (constraints satisfaction by the approximated solutions), convergence and optimality of the above algorithm.

Remark 2.2: i) For large-scale systems and networks the above algorithm induces a large communication overhead for exchanging information between distributed decision makers. In order to overcome this drawback, the authors of [6] proposed a two-level communication architecture and a three-step algorithm including an extra outer iterate step. To complete their work a proof feasibility, convergence and optimality of this three-step algorithm is documented in [7]. A simple algorithm for decomposing large problems into smaller ones via 'neighborhoods' is also documented there. ii) In this paper we are not concerned with communication overhead as it can be managed by decomposing the system into disjoint neighborhoods and using the three-step algorithm.

In order to analyze the computation time of the above consensus based distributed optimization algorithm, throughout this paper we use the following stopping criterion.

Definition 2.3: For a given $\epsilon > 0$, the two-step algorithm of [2] is terminated as soon as the following inequality holds

$$|J(u_1^t, ..., u_n^t) - J(u_1^{t-1}, ..., u_n^{t-1})| \leq \epsilon.$$

Note that for small values for ϵ, there will be very small improvement in the approximation of the optimal solution by the distributed optimization algorithm of [2]; and as the algorithm converges, the algorithm can therefore be terminated as soon as the above inequality holds.

Definition 2.4: (Total Number of Iterations for ϵ-Convergence) For a given $\epsilon > 0$, let T_ϵ be the smallest integer such that $|J(u_1^t, ..., u_n^t) - J(u_1^{t-1}, ..., u_n^{t-1})| \leq \epsilon$, $\forall t \geq T_\epsilon$. Then, T_ϵ is referred as the total number of iterations for ϵ-convergence.

We refer to $J(u_1^{T_\epsilon}, ..., u_n^{T_\epsilon})$ as an approximation of the optimal cost and the sequence $(u_1^{T_\epsilon}, ..., u_n^{T_\epsilon})$ as an approximation of the optimal solution.

III. COMPUTATION TIME ANALYSIS

The interior point method [1], [8] and the active set method [8], [9] are the most commonly used approaches for solving general QP problems. As a benchmark, we employ a generic interior point method [8] to solve the QP problem (2) using

the centralized optimization algorithm and the distributed algorithm of [2]. Computation time analysis of the active set method is investigated in [10].

Let n_c and n_d be the numbers of inequality constraints and decision variables of a QP problem, respectively. As stated in [8] at each iteration of the interior point method, the computing device must solve n_c times a system of n_d linear equations. Therefore, if the commonly used techniques, such as the generic Gauss-Jordan elimination technique or Gaussian elimination technique (which have cubic computational complexity) is used to solve this system of linear equations, the optimization computational complexity (i.e., the number of floating point arithmetic operations required to find the optimal solution) associated with each iteration of the interior point method is $\mathcal{O}(n_c \times n_d^3)$. As pointed out in [8], the number of iterations required for convergence of the interior point method to the optimal solution is insensitive to n_c and n_d. Therefore, the optimization computational complexity of the Interior Point Method (IPM) for solving QPs is

$$IPM \sim \mathcal{O}(n_c \times n_d^3). \tag{5}$$

In this paper we are concerned with computation time which is the empirical time spent by computing devices to find the solution. The computation time for finding the optimal solution consists of two terms: (i) Optimization time C_{opt}, which is the empirical time spent by computing devices to solve the optimization problem. This complexity term is proportional to the empirical number of floating point arithmetic operations required for solving the optimization problem. Therefore, if IPM is used, then

$$C_{opt} \sim \mathcal{O}(n_c \times n_d^3). \tag{6}$$

(ii) Constraint making time complexity term C_{cm} which is the empirical time spent for making constraints to be implemented in optimizer. This complexity term is also proportional to the empirical number of required floating point arithmetic operations.

In this section, the computation time (as described above) for the centralized algorithm and distributed algorithm are analyzed for the QP problem (2) for two cases: (i) Fixed N. (ii) Fixed n.

A. Fixed N, Varying n

In this section it is assumed that the horizon length N is fixed but the number of sub-systems n varies. It is also assumed that the distributed decision makers use the interior point method for their smaller QPs, and at each inner iteration updated decision variables are exchanged only when all optimizers finish their computation. That is, the distributed algorithm uses synchronized communication. Under these assumptions expressions for the computation times of the centralized algorithm and distributed algorithm in terms of the number of sub-systems n are presented in this section.

As the centralized optimization algorithm applied to the QP problem (2) involves $n_c = 6nN$ inequality constraints and $n_d = nN$ decision variables, from (6) it follows that the optimization time of the centralized algorithm is of order 4, i.e.,

$$C_{opt}(n) \sim \mathcal{O}(n_c \times n_d^3) \sim \mathcal{O}(n^4).$$

In addition, as the number of inequality constraints is a linear function of n, the complexity term C_{cm} is a linear function of n, i.e., $C_{cm}(n) \sim \mathcal{O}(n_c) \sim \mathcal{O}(n)$. Hence, the computation time of the centralized algorithm is of order 4, i.e.,

$$C_{cen}(n) = C_{opt}(n) + C_{cm}(n) \sim \mathcal{O}(n^4). \tag{7}$$

When the distributed optimization algorithm of [2] is applied to the QP problem (2), each decision maker i has a decision variable u_i of dimension N and the horizon length N determines its number of inequality constraints. Under the assumption of synchronized communication, the computation time of the distributed optimization algorithm of [2] in terms of the number of sub-systems, i.e., $C_{dis}(n)$, is given by

$$C_{dis}(n) = \sum_{j=1}^{T_\epsilon(n)} C_j(n), \tag{8}$$

where $T_\epsilon(n)$ is the empirical number of required iterations for ϵ-convergence and $C_j(n)$ is the computation time of the decision maker with the largest computation time at iteration j. This decision maker is referred to here as 'dominant decision maker'. As it is assumed that the distributed decision makers also use the interior point method for their smaller QPs, their optimization times are determined only by their number of inequality constraints, as each of them has N decision variables.

The horizon length N determines the number of inequality constraints affecting each decision maker. Specifically, if $n < \lceil \frac{N}{2} \rceil$, then the effect of a change in the decision variables of each sub-system is seen in all upstream sub-systems. Therefore, for $n < \lceil \frac{N}{2} \rceil$ the number of inequality constraints required to describe the QP associated with the sub-system at the end of the cascade (i.e. sub-system S_n) is $n_c = 2N(2n+1)$, which is the largest number of inequality constraints, compared to all other sub-systems. On the other hand, if $n \geq \lceil \frac{N}{2} \rceil$, then the effect of a change in the decision variables of each sub-system is seen in only a subset of the upstream sub-systems. Specifically, it is seen in the observation signals of $\lceil \frac{N}{2} \rceil$ upstream sub-systems and in the control signals of $\lfloor \frac{N}{2} \rfloor$ upstream sub-systems. This follows from the cascade topology of the system (1). Consequently, for $n \geq \lceil \frac{N}{2} \rceil$, the number $2N(2\lfloor \frac{N}{2} \rfloor + 1)$ (for an even N) or $2N(\lceil \frac{N}{2} \rceil + \lfloor \frac{N}{2} \rfloor + 1)$ (for an odd N) of inequality constraints in the QPs associated with the last few sub-systems remains unchanged, even if the number of upstream sub-systems increases. That is, several downstream decision makers have the same largest number of inequality constraints. Either way, the optimization time of the sub-system at the end of the cascade dominates. Similarly, the constraint making complexity term of the sub-system at the end of the cascade dominates. Therefore, the computation time of the last sub-system S_n dominates. That is, S_n is the dominant sub-system. Note that at each iteration j, the

dominant computation time consists of two terms: $C_j(n) = C_{opt,j}(n) + C_{cm,j}(n)$, where $C_{opt,j}(n)$ is the optimization time and $C_{cm,j}(n)$ is the constraint making time complexity term of the dominant sub-system (i.e., sub-system S_n) at iteration j.

From above, for a given number of sub-systems n, the dominant sub-system remains constant for all iterations, whereby the dominant computation time $C_j(n)$ also remains constant for $j > 1$, since there is no change in the number of inequality constraints associated with the dominant sub-system for $j \geq 1$. Therefore, $C_j(n) \doteq C(n) = C_{opt}(n) + C_{cm}(n), \forall j > 1$, where $C_{opt}(n)$ and $C_{cm}(n)$ are the optimization time and constraint making complexity term of the dominant sub-system. However, for $j = 1$, it takes some time that variables to be placed into the cache memory and therefore

$$C_1(n) \geq C_j(n) = C(n), \forall j > 1.$$

Hence, applying (8) to the QP problem (2) results in the following expression for the computation time.

$$C_{dis}(n) = C_1(n) + (T_\epsilon(n) - 1)C(n). \tag{9}$$

Remark 3.1: The formula (9) is applicable to any unstructured system having an unchanged dominant sub-system with computation time $C(n)$.

In summary, the maximum number of the inequality constraints affecting any sub-system is

$$n_c = \begin{cases} 2N(2\left\lfloor \frac{N}{2} \right\rfloor + 1), & \text{if} n \geq \left\lceil \frac{N}{2} \right\rceil \text{and } N \text{ is even,} \\ 2N(\left\lceil \frac{N}{2} \right\rceil + \left\lfloor \frac{N}{2} \right\rfloor + 1), & \text{if } n \geq \left\lceil \frac{N}{2} \right\rceil \text{,and } N \text{ is odd,} \\ 2N(2n+1), & \text{if } n < \left\lceil \frac{N}{2} \right\rceil, \end{cases}$$

and the number of decision variables of each sub-system is $n_d = N$. Hence, for the case of $n \geq \left\lceil \frac{N}{2} \right\rceil$, we have: $C_{opt}(n) \sim \mathcal{O}(n_c \times n_d^3) \sim \mathcal{O}(n^0)$, $C_{cm}(n) \sim \mathcal{O}(n_c) \sim \mathcal{O}(n^0)$, $C(n) = C_{opt}(n) + C_{cm}(n) \sim \mathcal{O}(n^0)$, and $C_1(n) \sim \mathcal{O}(n_c + n_d) \sim \mathcal{O}(n^0)$. Note that for the other case of $n < \left\lceil \frac{N}{2} \right\rceil$, $C_{opt}(n)$, $C_{cm}(n)$, $C(n)$ and $C_1(n)$ are linear functions of n. In addition, from the experimental results (given in the next section) it is observed that $T_\epsilon(n)$ as a function of n is approximated and upper bounded by a linear function. Hence, for the QP problem (2), from (9) it follows for the case of $n \geq \left\lceil \frac{N}{2} \right\rceil$ that $C_{dis}(n)$ has the following expression.

$$C_{dis}(n) \sim \mathcal{O}(n). \tag{10}$$

Note that for the other case of $n < \left\lceil \frac{N}{2} \right\rceil$, $C_{dis}(n)$ is a polynomial function of n with order 2.

Remark 3.2: Similar to the centralized optimization algorithm, the distributed optimization algorithm does not exploit the topology of the network either in its formulation or in the solutions of smaller QPs. The topology of the network just helped us to determine the dominant sub-system to analyze the computation time of the distributed optimization algorithm on cascade systems.

B. Fixed n, Varying N

In this section it is assumed that the number of sub-systems n is fixed but the horizon length N varies. Similar to the previous section it is also assumed that the distributed decision makers use the interior point method and synchronized communication. Under these assumptions expressions for the computation times in terms of the horizon length N are presented in this section.

As the centralized optimization algorithm applied to the QP problem (2) involves $n_c = 6nN$ inequality constraints and $n_d = nN$ decision variables, from a similar argument as we used in previous section, it follows that $C_{opt}(N) \sim \mathcal{O}(N^4)$, and $C_{cm}(N) \sim \mathcal{O}(N)$. Hence, the computation time of the centralized optimization algorithm is of order 4, i.e.,

$$C_{cen}(N) = C_{opt}(N) + C_{cm}(N) \sim \mathcal{O}(N^4). \tag{11}$$

For the distributed algorithm following a similar argument as we used in the previous section it also follows for the case of $N > 2n$ that $C_{opt}(N) \sim \mathcal{O}(N^4)$, $C_{cm}(N) \sim \mathcal{O}(N)$, and $C_1(N) \sim \mathcal{O}(N)$. Note that for the other case of $N \leq 2n$, $C_{opt}(N)$, $C_{cm}(N)$ and $C_1(N)$ are polynomial functions of N with order 5,2, and 2, respectively. In addition, from the experimental results (given in the next section) it is observed that $T_\epsilon(N)$ grows linearly with N. Hence, for the case of $N > 2n$, we have:

$$\begin{aligned} C_{dis}(N) &= C_1(N) + (T_\epsilon(N) - 1)(C_{opt}(N) + C_{cm}(N)) \\ &\sim \mathcal{O}(N^5). \end{aligned} \tag{12}$$

Note that for the other case, $C_{dis}(N)$ is a polynomial function of N with order 6.

Remark 3.3: The communication overhead analysis is reported in ([10], Section III), in which it is shown for automated irrigation networks that the communication overhead can be a quadratic function of the number of sub-systems n.

IV. Simulation Results

In this section, the expressions for the computation time are verified for the automated East Goulburn irrigation district No. 12 with a total 42 sub-systems (pools operating under distributed distant-downstream PI control for water-level regulation), which is of the form (1). This network of heterogeneous sub-systems represents a typical irrigation system in Australia and other locations globally. The computation times of the centralized optimization algorithm and consensus based distributed optimization algorithm of [2] are compared with each other in this section for two cases: (i) Fixed N and (ii) Fixed n. For the first case the expressions for the computation time are verified by increasing the number of upstream sub-systems. In particular, the centralized optimization algorithm and the consensus based distributed optimization algorithm of [2] are applied to the last 6, 12, 18, 24, 30, 36, and 42 sub-systems of the automated East Goulburn irrigation district No. 12. For the other case, we fix n to be $n = 42$ and verify the expressions for computation time for $N = \{24, 44, 64, 74, 84\}$. Throughout, it is assumed that the last sub-system of this irrigation district is subject to

n	$C_{cm}(sec.)$	$C_{opt}(sec.)$	$C_{cen}(sec.)$
6	4.15	9.63	13.73
12	7.24	47.46	54.7
18	12.45	150.6	163.05
24	19.25	321.33	340.58
30	28.74	1014.08	1024.82
36	33.98	3243.1	3277.1
42	43.5	5340.3	5383.8

TABLE I

TRADE-OFFS BETWEEN n, $C_{cm}(n)$, $C_{opt}(n)$ AND $C_{cen}(n)$.

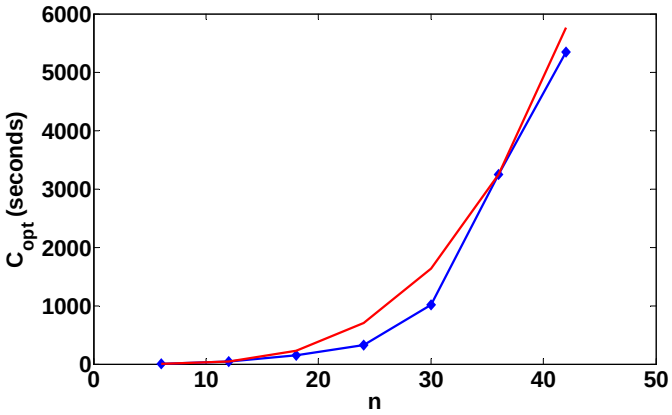

Fig. 1. Optimization time of the centralized algorithm versus the number of sub-systems n for $N = 84$. Blue dots are experimental data. Red curve includes the corresponding approximated data obtained by polynomial (13).

n	$C_1(sec.)$	$C_{cm}(sec.)$	$C_{opt}(sec.)$	T_ϵ	$C_{dis}(sec.)$
6	6.57	3.67	0.64	10	43.1
12	8.86	4.44	1.19	15	89.6
18	11.3	5.17	1.63	21	144.47
24	12.85	5.81	1.98	32	250.43
30	14.94	6.6	2.26	38	334.79
36	17.02	7.22	2.53	46	462.5
42	19.41	7.95	2.71	54	584.39

TABLE II

TRADE-OFFS BETWEEN n, $C_1(n)$, $C_{cm}(n)$, $C_{opt}(n)$, $T_\epsilon(n)$ AND $C_{dis}(n)$.

17.8041 m^3/min off-take disturbance and the water levels must be within $\pm 0.25m$ of the desired water levels. That is, the lower bounds on the water levels are set to be $L_i = r_i - 0.25$ and the upper bounds are set to be $H_i = r_i + 0.25$. It is also assumed that the upper bounds on the input flow rates are $Z_i = L_i^{\frac{3}{2}}$, the lower bounds are $E_i = 0$, $\pi_i = \frac{1}{n}$ and the weighting matrices Q, R, P in (2) are identity matrices. u_i^0 are chosen to be the desired water levels and ϵ is set to be 0.1. For optimization, the MATLAB solver *quadprog* is used, via YALMIP [11], to compute the solution to QPs numerically. The solver *quadprog* is set to use the interior point method [1],[8].

A. Fixed N, Varying n

In this section we fix the horizon length to be $N = 84$. Similar to [8] to find the computation time spent for computing the optimal solution for each $n = 6, 12, 18, 24, 30, 36, 42$, the simulation results are repeated several times by choosing different initial conditions: $x_i[0] = (\, g_i \quad 0 \quad . \quad . \quad . \quad 0\,)'$, where $g_i = r_i - 0.2$, $g_i = r_i - 0.18$, ... ,$g_i = r_i$, $g_i = r_i + 0.02$, ..., $g_i = r_i + 0.2$. Then, the average of the obtained times is calculated by excluding those results which are subject to infeasible optimization solution. This average represents the computation time for a given n.

Table I summarizes trade-offs between n, $C_{cm}(n)$, $C_{opt}(n)$ and $C_{cen}(n)$ for the centralized optimization algorithm and Fig. 1 and Fig. 2 show the optimization time and computation time of the centralized algorithm, respectively, applied to the QP problem (2). As clear from Fig. 1 and Fig. 2 the optimization time and computation time in terms of the number of sub-systems are approximated and upper bounded by the following functions, which are of order 4 (note that for $n > 6$, C_{cm} is approximated and upper bounded by the following linear function $C_{cm}(n) \approx 1.2087n - 7.264$).

$$C_{opt}(n) \approx \alpha_4 n^4 + \alpha_3 n^3 + \alpha_2 n^2 + 4.0212896n + 1.1336, \quad (13)$$

$$C_{cen}(n) = C_{opt}(n) + C_{cm}(n) \approx \alpha_4 n^4 + \alpha_3 n^3 + \alpha_2 n^2 + 5.2299896n - 6.1304. \quad (14)$$

Here $\alpha_4 = 0.001$, $\alpha_3 = 0.052134$ and $\alpha_2 = -0.7829$.

Remark 4.1: i) During the experiments it is observed that for a given n the overheads for different initial conditions are very close to the corresponding empirical value.
ii) The expressions for C_{cm} and C_{opt} are obtained by interpolating a linear function and a polynomial function of order 4, respectively. For C_{cm}, it is observed that there will be very small improvement in interpolation error if a higher order function is used. This is also true for C_{opt}. For C_{opt}, it is also observed that approximating C_{opt} by a lower order function results in a significant interpolation error.

Table II summarizes trade-offs between n, $C_1(n)$, $C_{cm}(n)$, $C_{opt}(n)$, $T_\epsilon(n)$ and the distributed algorithm computation time $C_{dis}(n)$. From this table it is observed that $C_1(n)$, $C_{cm}(n)$ and $T_\epsilon(n)$ are approximated and upper bounded by the following linear functions, respectively: $C_1(n) \approx 0.3379n + 5.3$, $C_{cm}(n) \approx 0.1192n + 3.025$, and $T_\epsilon(n) \approx 1.223n + 2.67$. From this table it also follows that the dominant optimization time $C_{opt}(n)$ is approximated and upper bounded by the following linear function: $C_{opt}(n) \approx 0.045n + 0.91$.

Consequently, from (9) it follows that $C_{dis}(n)$ is approximated as follows:

$$C_{dis}(n) \approx 0.2008n^2 + 5.4246n + 11.8714. \quad (15)$$

Fig. 3 compares the experimental data with the approximation given by the above function, which is obtained from the formula (9). From this figure it follows that the distributed algorithm computation time is approximated and upper bounded by the above quadratic function.

Fig. 4 compares the computation times of the centralized algorithm and the distributed optimization algorithm with each other for $N = 84$. From this figure it follows that for the irrigation network of interest to us there is a

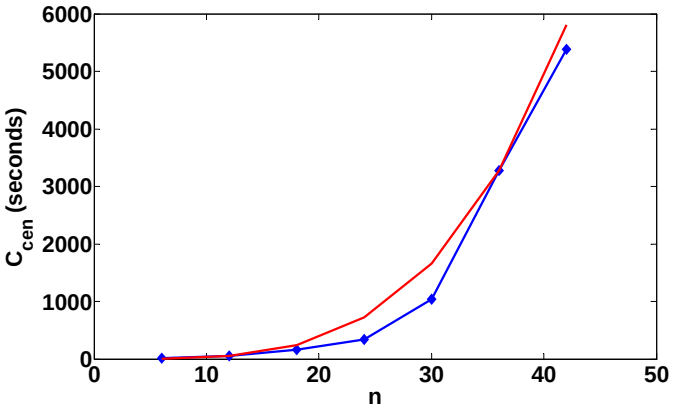

Fig. 2. Centralized computation time $C_{cen}(n)$ versus the number of sub-systems n for $N = 84$. Blue dots are experimental data. Red curve includes the corresponding approximated data obtained by polynomial (14).

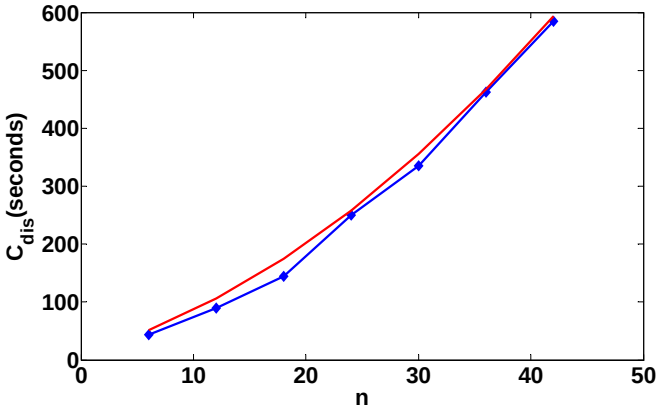

Fig. 3. $C_{dis}(n)$ versus the number of sub-systems for $N = 84$. Blue dots are experimental data. Red curve includes the corresponding approximated data obtained by polynomial (15).

significant advantage in terms of the computation time in using the distributed optimization algorithm. Specifically, the computation time of the centralized algorithm for $n = 42$ is $C_{cen}(42) = 5383.8$ seconds which is obviously intractable, while the computation time of the distributed optimization algorithm is $C_{dis}(42) = 584.39$ seconds. Also, when $nN > 2500$ there is a significant reduction in computation time if the distributed optimization algorithm is used.

B. Fixed n, Varying N

In this section we fix the number of sub-systems to be $n = 42$ and vary the horizon length N. Table III summarizes trade-offs between $N = \{24, 44, 64, 74, 84\}$, $C_{cm}(N)$, $C_{opt}(N)$ and $C_{cen}(N)$. From this table it follows that the above complexity terms are approximated and upper bounded by the following functions.

$$\begin{aligned} C_{cm}(N) &\approx 0.6372N - 9.172, \\ C_{opt}(N) &\approx \beta_4 N^4 + \beta_3 N^3 + \beta_2 N^2 \\ &\quad +9.488987N - 590.67916, \end{aligned}$$

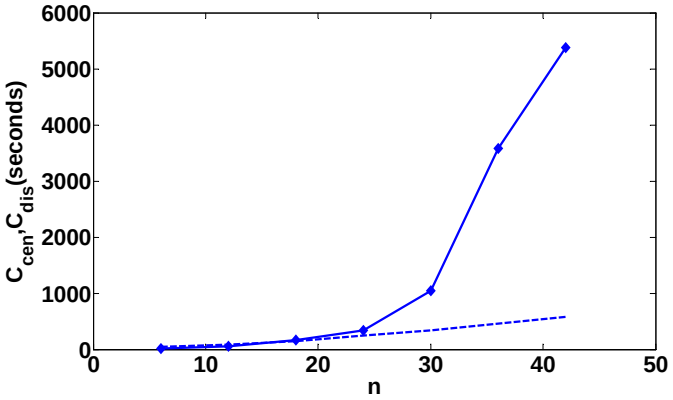

Fig. 4. $C_{cen}(n)$ and $C_{dis}(n)$ versus the number of sub-systems for $N = 84$. Solid curve indicates $C_{cen}(n)$ and dashed curve indicates $C_{dis}(n)$.

N	$C_{cm}(sec.)$	$C_{opt}(sec.)$	$C_{cen}(sec.)$
24	5.12	51.75	56.86
44	16.27	244.85	261.12
54	25	353.64	378.64
64	30.31	688.75	719.07
74	35.57	2201.43	2237
84	45.35	5025.9	5070.3

TABLE III
TRADE-OFFS BETWEEN N, $C_{cm}(N)$, $C_{opt}(N)$ AND $C_{cen}(N)$.

$$\begin{aligned} C_{cen}(N) &= C_{opt}(N) + C_{cm}(N) \\ &\approx \beta_4 N^4 + \beta_3 N^3 + \beta_2 N^2 \\ &\quad +10.126187N - 599.851161. \end{aligned} \tag{16}$$

Here $\beta_4 = 0.000615525$, $\beta_3 = -0.067118567$ and $\beta_2 = 1.97799323$.

Remark 4.2: During the experiments it is observed that for a given N the overheads of different initial conditions are very close to the corresponding empirical value.

Table IV summarizes trade-offs between N, $C_1(N)$, $C_{cm}(N)$, $C_{opt}(N)$, $T_\epsilon(N)$ and $C_{dis}(N)$. From this table it follows that the above complexity terms are approximated and upper bounded by the following functions.

$$C_1(N) \approx 0.004863N^2 - 0.231N + 4.52,$$

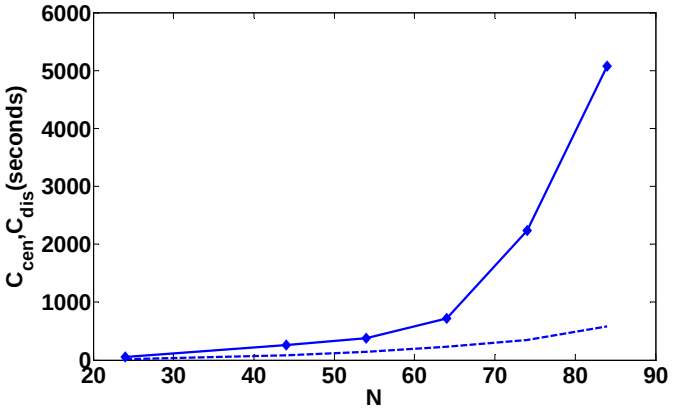

Fig. 5. $C_{cen}(N)$ and $C_{dis}(N)$ versus the horizon length N for $n = 42$. Solid curve indicates $C_{cen}(N)$ and dashed curve indicates $C_{dis}(N)$.

N	$C_1(sec.)$	$C_{cm}(sec.)$	$C_{opt}(sec.)$	T_ϵ	$C_{dis}(sec.)$
24	1.77	0.79	0.196	18	18.53
44	3.76	1.9	0.48	35	84.44
54	5.89	2.82	0.77	39	140.5
64	8.87	3.99	1.24	44	229.58
74	12.33	5.55	1.69	48	348.99
84	19.41	7.95	2.71	54	584.39

TABLE IV

TRADE-OFFS BETWEEN N, $C_1(N)$, $C_{cm}(N)$, $C_{opt}(N)$, $T_\epsilon(N)$ AND $C_{dis}(N)$.

$$\begin{aligned} C_{cm}(N) &\approx 0.001595N^2 - 0.053016N + 1.1432, \\ C_{opt}(N) &\approx \eta_5 N^5 + \eta_4 N^4 + \eta_3 N^3 + \eta_2 N^2 \\ &\quad + \eta_1 N + \eta_0, \\ T_\epsilon(N) &\approx 0.475N + 14.1, \\ C_{dis}(N) &= C_1(N) + (T_\epsilon(N) - 1)(C_{opt}(N) + C_{cm}(N)) \\ &\approx \gamma_6 N^6 + \gamma_5 N^5 + \gamma_4 N^4 + \gamma_3 N^3 \\ &\quad + \gamma_2 N^2 + \gamma_1 N + \gamma_0. \end{aligned} \quad (17)$$

Here $\eta_5 = 1.6703099 \times 10^{-8}$, $\eta_4 = 3.63306543 \times 10^{-6}$, $\eta_3 = 2.85851661451 \times 10^{-4}$, $\eta_2 = -9.1807458 \times 10^{-3}$, $\eta_1 = 0.108077$, $\eta_0 = 0.011009$, $\gamma_6 = 7.933972 \times 10^{-9}$, $\gamma_5 = -1.5068 \times 10^{-6}$, $\gamma_4 = 8.81864 \times 10^{-5}$, $\gamma_3 = 1.4183 \times 10^{-4}$, $\gamma_2 = -6.83463 \times 10^{-2}$, $\gamma_1 = 1.038541$ and $\gamma_0 = 19.64014$.

Fig. 5 compares the computation times of the centralized algorithm and the distributed optimization algorithm with each other for $n = 42$. From this figure it also follows that there is a significant advantage in terms of the computation time in using the distributed algorithm for the irrigation network of interest to us. That is, while the computation time of the distributed optimization algorithm is described by a polynomial function of order 6, on the basis of experiments for the particular system of interest to us, the growth of the distributed algorithm computation time in N is bounded above by that of the centralized algorithm. Specifically, when $nN > 2500$ there is a significant reduction in computation time if the distributed optimization algorithm is used. This follows as each decision maker frequently updates its local component of the overall decision variable by solving an optimization problem of reduced size.

V. CONCLUSIONS

In this paper we compared the computation time of a centralized optimization algorithm for solving the QP problem (2), with the computation time of the distributed optimization algorithm of [2]. It was assumed that both algorithms use the interior point method and do not exploit problem structure. For the QP problem (2), which represents the typical optimization problem arising in automated irrigation networks, it was illustrated that there is a significant advantage in terms of computation time in using the distributed optimization algorithm of [2] for large-scale networks. Specifically, for the particular network of interest to us (East Goulburn irrigation district No. 12) it was shown that the centralized optimization algorithm cannot provide a computationally tractable solution; and there is a significant reduction in the computation time when $nN > 2500$ if the distributed optimization algorithm of [2] is used.

REFERENCES

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[2] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, Cooperative distributed model predictive control, *Systems and Control Letters*, vol. 59, pp. 460-469, 2010.

[3] J. M. Maestre, D. Munoz de la Pena, E. F. Camacho, and T. Alamo, Distributed model predictive control based on agent negotiation, *Journal of Process Control*, vol. 21, pp. 685-697, 2011.

[4] R. R. Negenborn, *Multi-agent model predictive control with applications to power networks*, PhD Thesis, Delft University of Technology, 2007.

[5] R. Scattolini, Architecture for distributed and hierarchical model predictive control - a review, *Journal of Process Control*, 19(5), pp. 723-731, 2009.

[6] B. T. Stewart, J. B. Rawlings and S. J. Wright, Hierarchical cooperative distributed model predictive control, *Proceedings of the 2010 American Control Conference*, pp. 3963-3968, 2010.

[7] A. Farhadi, M. Cantoni, and P. M. Dower, Performance and information pattern trade-off in a consensus based distributed optimization method, *in Proceedings of the Second Australian Control Conference*, Sydney, Australia, pp. 373-380, November 2012.

[8] M. S. K. Lau, S. P. Yua, K. V. Ling, and J. M. Maciejowski, A comparison of interior point and active set methods for FPGA implementation of model predictive control, *in Proceedings of the European Control Conference*, pp. 157-161, 2009.

[9] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Sigma Series in Applied Mathematics, 1988.

[10] A. Farhadi, P. M. Dower and M. Cantoni, Computation time analysis of centralized and distributed optimization algorithms applied to automated irrigation networks, *in Proceedings of 2013 Australian Control Conference*, Perth, Australia, November 4-5, 2013.

[11] J. Lofberg, Yalmip: a toolbox for modeling and optimization in MATLAB, in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip