Computation Time Analysis of Centralized and Distributed Optimization Algorithms Applied to Automated Irrigation Networks

Alireza Farhadi, Peter M. Dower, and Michael Cantoni

***Abstract*— This paper compares 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 active set method that does not exploit problem structure. The second is distributed and is based on a consensus algorithm, not specifically tailored to account for system structure. It is shown that there is a significant advantage in terms of computation overhead (the time spent computing the optimal solution) in using the second algorithm in large-scale networks. Specifically, for a fixed horizon length the computation overhead of the centralized algorithm grows as $\mathcal{O}(n^5)$ with the number n of sub-systems. By contrast, it is observed via a combination of analysis and experiment that given n times resources for computation the computation overhead 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 (each representing a connected water pool) of the following form:

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

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:

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

where

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

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. Generic algorithms for centralized computation, such as the active set method [1], involving one centralized computation resource that is responsible for computing the controls u_i 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 overhead (i.e., the time spent computing the optimal solution) of the centralized optimization algorithms is not practical. Towards overcoming this computational scalability problem, a consensus based distributed optimization algorithm is proposed in [2] that exploits the computational power often available at distributed sub-systems/decision makers (for a brief description of this algorithm, see [3], Section II). This distributed algorithm can be used to approximate the solution of the structured optimal control problem (2).

B. PAPER CONTRIBUTIONS

This paper compares the computation overhead of the centralized optimization algorithm based on the active set method [1] with the computation overhead of the aforementioned consensus based distributed optimization algorithm of [2] for the QP problem (2) subject to inequality constraints. For a fixed horizon length, it is shown that the computation overhead of the centralized algorithm grows as $\mathcal{O}(n^5)$ with the number n of sub-systems. By contrast, it is observed via a combination of analysis and experiment that the computation overhead of the distributed algorithm grows as $\mathcal{O}(n)$ with

*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. P. M. Dower and M. Cantoni are with the Department of Electrical and Electronic Engineering, University of Melbourne, Victoria, Australia. E-mail: {pdower,cantoni}@unimelb.edu.au

the number n of sub-systems given n times resources for computation when the computational power of each distributed resource is the same as the centralized resource. Similarly, for a fixed number of sub-systems, it is shown that the computation overhead of the centralized algorithm and the distributed algorithm grows as $\mathcal{O}(N^5)$ and $\mathcal{O}(N^6)$ with the horizon length N, respectively. 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 overhead in N is bounded above by that of the centralized algorithm. In summary, empirical evidence suggests that there is a significant advantage in terms of the computation time in using the consensus based distributed optimization algorithm of [2] in large-scale irrigation networks.

C. PAPER ORGANIZATION

The paper is organized as follows: Section II is allocated to computation overhead analysis. This is followed by communication overhead analysis in Section III. Simulation results are presented in Section IV and the paper is concluded in Section V.

II. COMPUTATION OVERHEAD ANALYSIS

In this section we analyze the computation overhead of the aforementioned centralized and consensus based distributed optimization algorithms. Throughout, we use the following stopping criterion to approximate the optimal solution of the QP problem (2) by the distributed optimization algorithm.

Definition 2.1: For a given $\epsilon > 0$, the distributed 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 [2], the algorithm can be therefore terminated as soon as the above inequality holds.

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.

The formal definition of computation overhead is given below. It involves optimization time and constraint making time complexity term as defined as follows:

Definition 2.2: (Optimization Time): Optimization time C_{opt} is the empirical time spent by optimizers/decision makers to approximate the optimal solution.

Definition 2.3: (Constraint Making Time Complexity): Constraint making time complexity term C_{cm} is the empirical time spent for making constraints to be implemented in optimizers.

Definition 2.4: (Computation Overhead): Computation overhead denoted by C_{cen} for the centralized algorithm and C_{dis} for the distributed algorithm is C_{opt} plus C_{cm}.

We have one more definition.

Definition 2.5: (Optimization Computational Complexity): Optimization computational complexity is the empirical number of floating point arithmetic operations (addition, multiplication, devision) to be executed to find the optimal solution.

The active set method [1], [4] and the interior point method [4], [5] are the most commonly used approaches for solving general QP problems. As a benchmark, we employ a generic active set method [4] to solve the QP problem (2) using the centralized optimization algorithm and the distributed algorithm of [2]. Computation overhead analysis of the interior point method is reported in [3].

Let n_c and n_d be the numbers of inequality constraints and decision variables of a QP problem, respectively. As stated in [4] at each iteration of the Active Set Method (ASM), the computing device must solve a system of linear equations with empirical number of $n_d + \frac{n_c}{2}$ linear equations (i.e., the empirical number of linear equations$\sim \mathcal{O}(n_d + \frac{n_c}{2})$). 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 associated with each iteration of the active set method is $\mathcal{O}\Big((n_d+\frac{n_c}{2})^3\Big)$. As shown in [4], the empirical number of iterations required for the convergence of the active set method to the optimal solution grows as $\mathcal{O}(n_d n_c)$. Therefore, the optimization computational complexity of ASM for solving QPs is

$$ASM \sim \mathcal{O}\Big(n_d n_c (n_d + \frac{n_c}{2})^3\Big). \tag{4}$$

The optimization time C_{opt} is proportional to the optimization computational complexity. Therefore, if ASM is used, then

$$C_{opt} \sim \mathcal{O}\Big(n_d n_c (n_d + \frac{n_c}{2})^3\Big). \tag{5}$$

In this section, the computation overhead for the centralized and distributed algorithms 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 active set 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 overheads 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 (5) it follows that the optimization time of the centralized algorithm is of order 5,

i.e.,

$$C_{opt}(n) \sim \mathcal{O}\Big(n_d n_c (n_d + \frac{n_c}{2})^3\Big) \sim \mathcal{O}(n^5).$$

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 overhead of the centralized algorithm is of order 5, i.e.,

$$C_{cen}(n) = C_{opt}(n) + C_{cm}(n) \sim \mathcal{O}(n^5). \tag{6}$$

When the distributed optimization algorithm of [2] is applied to the QP problem (2), each decision maker/sub-system 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 overhead of the distributed optimization algorithm of [2] in terms of the number of sub-systems, i.e., $C_{dis}(n)$, is given by

$$C_{dis}(n) = \sum_{j=1}^{T_\epsilon(n)} \mathcal{C}_j(n), \tag{7}$$

where $T_\epsilon(n)$ is the empirical number of required iterations for ϵ-convergence and $C_j(n)$ is the computation overhead of the decision maker with the largest computation overhead at iteration j. This decision maker is referred to here as 'dominant decision maker'. As shown in [3] the computation overhead of the last sub-system S_n dominates as it has the largest number of inequality constraints. Note that at each iteration j, the dominant computation overhead (overhead of sub-system S_n) 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.

For a given number of sub-systems n, the dominant sub-system remains the same for all iterations, whereby the dominant computation overhead $C_j(n)$ also remains the same for $j > 1$, since there is no change in the number of decision variables and 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 (7) to the QP problem (2) results in the following expression for the computation overhead.

$$C_{dis}(n) = \mathcal{C}_1(n) + (T_\epsilon(n) - 1)C(n). \tag{8}$$

As shown in [3], the number of the inequality constraints of the dominant sub-system is

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

and the number of decision variables of each sub-system is $n_d = N$. Hence, for the case of $n \geq \lceil \frac{N}{2} \rceil$, we have: $C_{opt}(n) \sim \mathcal{O}\Big(n_d n_c (n_d + \frac{n_c}{2})^3\Big) \sim \mathcal{O}(n^0)$, $C_{cm}(n) \sim \mathcal{O}(n_c) \sim \mathcal{O}(n^0)$, $C(n) = C_{opt}(n) + C_{cm}(n) \sim \mathcal{O}(n^0)$, and $C_1(n) \sim \mathcal{O}(n_c + n_d) \sim \mathcal{O}(n^0)$. Note that for the other case of $n < \lceil \frac{N}{2} \rceil$, $C_{opt}(n)$, $C_{cm}(n)$, $C(n)$ and $C_1(n)$ are polynomial functions of n with order 4,1,4,1, respectively. In addition, from the experimental results (given in Section IV) 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 (8) it follows for $n \geq \lceil \frac{N}{2} \rceil$ that $C_{dis}(n)$ has the following expression.

$$C_{dis}(n) \sim \mathcal{O}(n). \tag{9}$$

Note that for the other case, $C_{dis}(n)$ is a polynomial function of n with order 5.

Remark 2.6: 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 overhead 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 active set method and synchronized communication. Under these assumptions expressions for the computation overheads 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, following a similar argument to the previous section, it follows that

$$C_{opt}(N) \sim \mathcal{O}(N^5), \quad C_{cm}(N) \sim \mathcal{O}(N).$$

Hence, the computation overhead of the centralized optimization algorithm is of order 5, i.e.,

$$C_{cen}(N) = C_{opt}(N) + C_{cm}(N) \sim \mathcal{O}(N^5). \tag{10}$$

Similarly, for the distributed algorithm it follows for the case of $N > 2n$ that $C_{opt}(N) \sim \mathcal{O}(N^5)$, $C_{cm}(N) \sim \mathcal{O}(N)$, 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 9, 2 and 2, respectively. In addition, from the experimental results (given in Section IV) it is observed that $T_\epsilon(N) \sim \mathcal{O}(N)$. Therefore, for $N > 2n$, we have

$$\begin{aligned} C_{dis}(N) &= C_1(N) + (T_\epsilon(N) - 1)(C_{opt}(N) + C_{cm}(N)) \\ &\sim \mathcal{O}(N^6). \end{aligned} \tag{11}$$

Note that for the other case, $C_{dis}(N)$ is a polynomial function of N with order 10.

Remark 2.7: Experimental results (given in Section IV) for the case of $N \leq 2n$ reveals that for a small N (e.g.,

$N \leq 24$) the optimization time $C_{opt}(N)$ is approximated by a linear function; and consequently, $C_{dis}(N)$ is a polynomial function of N with order 3.

III. COMMUNICATION OVERHEAD ANALYSIS

In this section we are concerned with communication overhead as defined below.

Definition 3.1: (Communication Overhead): Communication overhead C_{com} is the total time spent for exchanging information between sub-systems/decision makers to have an approximation of the optimal solution up to an apriori fixed precision ϵ.

Note that for the centralized algorithm the communication overhead is zero as this algorithm involves only one centralized computation resource. Consequently, the computation time of the centralized algorithm is $C^{cen}_{total} \doteq C_{cen}$, while the computation time of the distributed algorithm is $C^{dis}_{total} \doteq C_{dis} + C_{com}$.

Throughout this section it is assumed that the irrigation network consists of heterogeneous pools with different lengths p_i, $i = 1, 2, ..., n$ distributed along a straight line, communication is wireless with the bandwidth of BW between each two sub-systems, and the communication network protocol is Time Division Multiple Access (TDMA) scheme. That is, a specific time slot is allocated to each subsystem to broadcast its information while other sub-systems are waiting to receive the transmitted information. It is also assumed that the communication topology is described by an undirected connected graph. The communication range of each sub-system is denoted by d, while the effective range under which a reliable communication is guaranteed is αd where $0 < \alpha \leq 1$. Each variable (decision variable and initial water level) is also encoded into a string of binaries with length l for transmission.

Under the above assumptions let L denote the distance between sub-system/pool S_1 and pool S_n gates. For $L \leq \alpha d$, a time slot of $s = \frac{2l}{BW}$ is allocated to each sub-system when the distributed algorithm is initialized (i.e., at $t = 0$) to broadcast its chosen decision variable u^0_i and its measured water level at $t = 0$ to all other sub-systems. For the rest of iterations $t \in \{1, 2, .., T_\epsilon\}$ a time slot of $s = \frac{l}{BW}$ is allocated. Hence, the communication overhead for the case of $L \leq \alpha d$ is

$$C_{com} = \frac{2nl}{BW} + \frac{nl}{BW} T_\epsilon = \frac{nl}{BW}(2 + T_\epsilon), \tag{12}$$

where T_ϵ is the number of iterations for ϵ-convergence.

Now, consider the other case of $L > \alpha d$. As the communication range of each sub-system is limited, the irrigation network must be decomposed into at least $q \doteq \lceil \frac{L}{\alpha d} \rceil$ disjoint neighborhoods of sub-systems to have a connected communication graph via multi-hopping. For the simplicity, suppose that sub-systems S_1 and S_n are contained in neighborhoods $\mathcal{N}_1$ and $\mathcal{N}_q$, respectively. The intermediate neighborhoods are denoted by $\mathcal{N}_2$,..., $\mathcal{N}_{q-1}$. For neighborhoods $\mathcal{N}_1$ and $\mathcal{N}_q$, one sub-system is chosen as cluster head and for intermediate neighborhoods two sub-systems are chosen as clusters heads. In each neighborhood, clusters heads are chosen such that they are within the effective communication range of their closest neighborhood sub-systems.

Now, consider the distributed algorithm of [2]. At each iteration, for exchanging information between all sub-systems, first an intra-neighborhood communication takes place by allocating a time slot of $s = \frac{Ml}{BW}$ seconds to each subsystem where $M = 2$ when the algorithm is initialized and $M = 1$ for $t \geq 1$. This is followed by communication between clusters heads acting as relays to exchange information between neighborhoods and subsequently between all sub-systems. For communication between neighborhoods a TDMA scheme is used by allocating a time slot of $\bar{s} = \frac{Mm_{max}l}{BW}$ to each cluster head where m_{max} is the number of sub-systems in the biggest neighborhood. Consequently, at each iteration it takes $M(q-1)\frac{m_{max}l}{BW}$ seconds to transfer information from neighborhood $\mathcal{N}_1$ to all of its downstream neighborhoods/sub-systems, and $M(q-2)\frac{m_{max}l}{BW}$ seconds to transfer information from neighborhood $\mathcal{N}_2$ to all of its downstream neighborhoods/sub-systems, etc. Similarly, it takes $M(q-1)\frac{m_{max}l}{BW}$ seconds to transfer information from neighborhood $\mathcal{N}_q$ to all of its upstream neighborhoods/subsystems, and $M(q-2)\frac{m_{max}l}{BW}$ seconds to transfer information from neighborhood $\mathcal{N}_{q-1}$ to all of its upstream neighborhoods/sub-systems, etc. This results in the total overhead of $\frac{Mnl}{BW} + 2\Big((q-1) + (q-2) + (q-3) + ... + 2 + 1\Big)\frac{Mm_{max}l}{BW} = \frac{Mnl}{BW} + q(q-1)\frac{Mm_{max}l}{BW}$ for exchanging information between all neighborhoods; and therefore, between all sub-systems at each iteration. Hence, the communication overhead for this case is

$$C_{com} = \left(\frac{nl}{BW} + \frac{q(q-1)m_{max}l}{BW}\right)(2 + T_\epsilon). \tag{13}$$

In summary, we have the following expression for the communication overhead:

$$C_{com} = \begin{cases} \frac{nl}{BW}(2 + T_\epsilon), & \text{if } L \leq \alpha d, \\ \left(\frac{nl}{BW} + \frac{q(q-1)m_{max}l}{BW}\right)(2 + T_\epsilon), & \text{if } L > \alpha d. \end{cases}$$

IV. SIMULATION RESULTS

In this section, the expressions for the computation overhead 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 overheads 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 overhead 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 subsystems of the automated East Goulburn irrigation district

No. 12. For the other case, we fix n to be $n = 42$ and we verify the expressions for computation overhead for $N = \{6, 9, 12, 15, 18, 20, 21, 22, 23, 24\}$. Throughout, it is assumed that the last sub-system of this irrigation district is subject to 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. The total length of this irrigation district is $L = 79590$m and the communication parameters are BW= 25Kbits, $l = 32$bits, $d = 15000$m and $\alpha = \frac{2}{3}$. For communication, this irrigation network is divided into 8 heterogeneous neighborhoods, as follows: $\mathcal{N}_1 = \{S_1, .., S_9\}$, $\mathcal{N}_2 = \{S_{10}, ..., S_{14}\}$, $\mathcal{N}_3 = \{S_{15}, S_{16}, S_{17}\}$, $\mathcal{N}_4 = \{S_{18}, ..., S_{21}\}$, $\mathcal{N}_5 = \{S_{22}, S_{23}, S_{24}\}$, $\mathcal{N}_6 = \{S_{25}, ..., S_{28}\}$, $\mathcal{N}_7 = \{S_{29}, ..., S_{35}\}$ and $\mathcal{N}_8 = \{S_{36}, ..., S_{42}\}$. Note that sub-systems S_9, S_{10}, S_{14}, S_{15}, S_{17}, S_{18}, S_{21}, S_{22}, S_{24},S_{25}, S_{28}, S_{29}, S_{35} and S_{36} are chosen as clusters heads. All of the sub-systems of neighborhood $\mathcal{N}_2$ are in the effective communication range of cluster head S_9. Similarly, all of the sub-systems of neighborhood $\mathcal{N}_1$ are in the effective range of cluster head S_{10} and all of the sub-systems of neighborhood $\mathcal{N}_3$ are in the effective range of cluster head S_{14}, and so on and so for. For optimization, the MATLAB solver *quadprog* is used, via YALMIP [6], to compute the solution to QPs numerically. The solver *quadprog* is set to use the active set method [4].

A. Fixed N, Varying n

In this section we fix the horizon length to be $N = 24$. Similar to [4] to find the computation overhead 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 overheads is calculated by excluding those results which are subject to infeasible optimization solution. This average represents the computation overhead 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 overhead of the centralized algorithm, respectively, applied to the QP problem (2). As clear from Fig. 1 and Fig. 2 the optimization time and computation overhead in terms of the number of sub-systems are approximated and upper bounded by the following polynomials, which are of order 5 (note that for $n > 12$, C_{cm} is approximated and upper bounded by the following linear function $C_{cm}(n) \approx 0.1308n - 0.375$).

$$\begin{aligned} C_{opt}(n) &\approx \alpha_5 n^5 + \alpha_4 n^4 + \alpha_3 n^3 + \alpha_2 n^2 \\ &\quad + 1886.667n - 5203.66, \end{aligned} \tag{14}$$

n	$C_{cm}(sec.)$	$C_{opt}(sec.)$	$C_{cen}(sec.)$
6	1.18	15.24	16.42
12	1.58	149.36	150.94
18	1.98	925.81	927.79
24	-	-	-
30	3.42	6577.21	6580.64
36	4.26	9496.95	9501.22
42	5.12	14034.94	14040.06

TABLE I

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

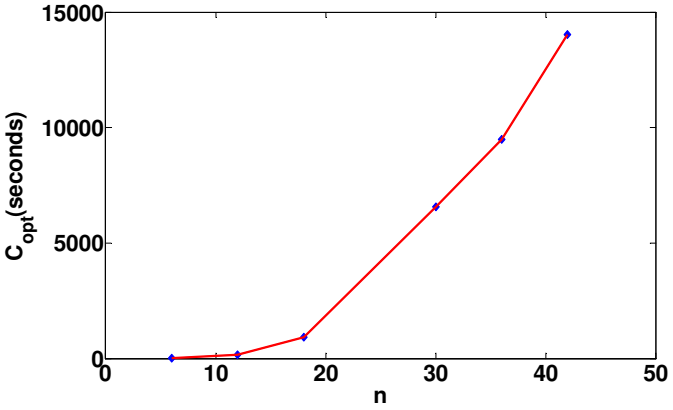

Fig. 1. Optimization time of the centralized algorithm versus the number of sub-systems n for $N = 24$. Blue dots are experimental data. Red curve includes the corresponding approximated data obtained by the polynomial (14).

$$\begin{aligned} C_{cen}(n) &= C_{opt}(n) + C_{cm}(n) \\ &\approx \alpha_5 n^5 + \alpha_4 n^4 + \alpha_3 n^3 + \alpha_2 n^2 \\ &\quad + 1886.7978n - 5204.035. \end{aligned} \tag{15}$$

Here $\alpha_5 = 0.002618280607$, $\alpha_4 = -0.306171682097$, $\alpha_3 = 13.0855694$ and $\alpha_2 = -237.504$.

Remark 4.1: i) During the experiments for $n = 24$ it is observed that for non of initial states the active set method converges to the optimal solution.

ii) The expressions for C_{cm} and C_{opt} are obtained by interpolating a linear function and a polynomial function of order 5, 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 overhead $C_{dis}(n)$. From this table it is observed that $C_1(n)$, $C_{cm}(n)$ and $T_\epsilon(n)$ are approximated (except for small $n = 6$) and upper bounded by the following functions, respectively: $C_1(n) \approx 1.77$, $C_{cm}(n) \approx 0.79$, and $T_\epsilon(n) \approx 0.334n + 4$. From this table it also follows that the dominant optimization time $C_{opt}(n)$ is approximated and upper bounded by the following: $C_{opt}(n) \approx 0.164$.

Consequently, from (8) it follows that $C_{dis}(n)$ is approximated as follows: $C_{dis}(n) \approx 0.334n + 4$.

Remark 4.2: Although the centralized algorithm is unable to find the optimal solution for $n = 24$, the distributed algorithm is able to approximate the optimal solution for

n	$C_1(sec.)$	$C_{cm}(sec.)$	$C_{opt}(sec.)$	T_ϵ	$C_{dis}(sec.)$
6	1.56	0.66	0.11	4	3.65
12	1.63	0.76	0.16	8	7.52
18	1.6	0.76	0.164	10	9.18
24	1.64	0.75	0.163	11	10.35
30	1.73	0.78	0.164	12	11.42
36	1.76	0.77	0.164	16	15
42	1.77	0.79	0.164	18	17.64

TABLE II

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

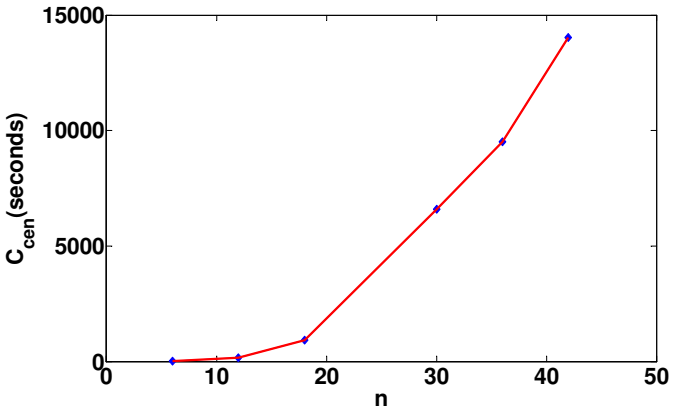

Fig. 2. Centralized computation overhead $C_{cen}(n)$ versus the number of sub-systems n for $N = 24$. Blue dots are experimental data. Red curve includes the corresponding approximated data obtained by polynomial (15).

this case.

Fig. 3 compares the computation overheads of the centralized algorithm and the distributed optimization algorithm with each other for fixed $N = 24$. From this figure it follows that for the irrigation network of interest to us there is a significant advantage in terms of the computation overhead in using the distributed optimization algorithm. Specifically, the computation overhead of the centralized algorithm for $n = 42$ is $C_{cen}(42) = 14040.06$ seconds which is obviously intractable, while the computation overhead of the distributed optimization algorithm is $C_{dis}(42) = 17.64$ seconds. Also, when $nN > 400$ there is a significant reduction in computation overhead if the distributed optimization algorithm is used.

Remark 4.3: Under the aforementioned assumptions for communication, we have $C_{com}(42) = 13.98$ seconds and therefore, the computation time of the distributed algorithm is $C^{dis}_{total}(42) = C_{dis}(42) + C_{com}(42) = 31.62$ seconds.

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 $N = \{6, 9, 12, 15, 18, 24\}$, $C_{cm}(N)$, $C_{opt}(N)$ and $C_{cen}(N)$. From this table it follows that the above complexity terms are approximated and upper bounded (except for the small $N = 6$) by the following polynomials.

$$\begin{aligned} C_{cm}(N) &\approx 0.236N - 0.544, \\ C_{opt}(N) &\approx \beta_5 N^5 + \beta_4 N^4 + \beta_3 N^3 + \beta_2 N^2 \\ &\quad +2203.9943N - 5490, \\ C_{cen}(N) &= C_{opt}(N) + C_{cm}(N) \\ &\approx \beta_5 N^5 + \beta_4 N^4 + \beta_3 N^3 + \beta_2 N^2 \\ &\quad +2204.2303N - 5490.544. \end{aligned}$$

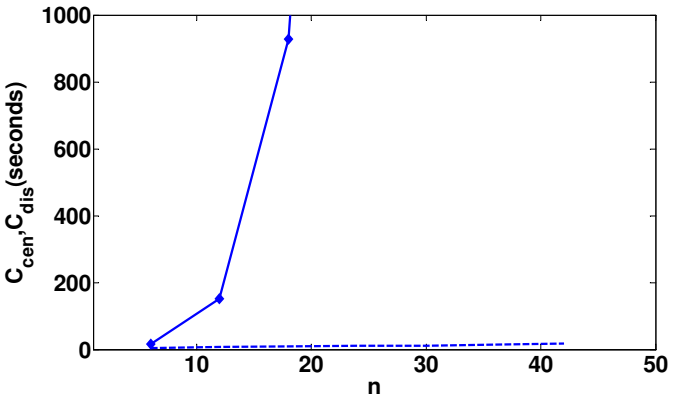

Fig. 3. $C_{cen}(n)$ and $C_{dis}(n)$ versus the number of sub-systems for $N = 24$. Solid curve indicates $C_{cen}(n)$ and dashed curve indicates $C_{dis}(n)$.

N	$C_{cm}(sec.)$	$C_{opt}(sec.)$	$C_{cen}(sec.)$
6	1.01	67.91	68.92
9	1.58	172.68	174.26
12	2.07	493.81	495.88
18	2.94	1606.3	1609.2
20	3.27	4729.9	4733.2
24	5.12	14034.94	14040.06

TABLE III

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

Here $\beta_5 = 0.001$, $\beta_4 = -0.30721$, $\beta_3 = 17.064$ and $\beta_2 = -304.32681$. 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 (except for small $N = 6, 9$) by the following polynomials.

$$\begin{aligned} C_1(N) &\approx 0.00075N^2 - 0.001N + 1.427, \\ C_{cm}(N) &\approx 0.0003542N^2 + 0.0219584N + 0.0655, \\ C_{opt}(N) &\approx 0.0055N + 0.032, \\ T_\epsilon(N) &\approx 1.4N - 14.7, \\ C_{dis}(N) &= C_1(N) + (T_\epsilon(N) - 1)(C_{opt}(N) + C_{cm}(N)) \\ &\approx 0.00049584N^3 + 0.03364N^2 - 0.2955N \\ &\quad -0.10375. \end{aligned} \tag{16}$$

Fig. 4 compares the experimental data with the approximation given by the above polynomial for $C_{dis}(N)$, which is obtained from the formula (8). From this figure it follows that the distributed algorithm computation overhead is approximated and upper bounded (except for small $N = 6, 12$) by the above cubic function.

Fig. 5 compares the computation overheads of the centralized algorithm and the distributed optimization algorithm with each other for $n = 42$. From this figure it also

N	$C_1(sec.)$	$C_{cm}(sec.)$	$C_{opt}(sec.)$	T_ϵ	$C_{dis}(sec.)$
6	1.39	0.21	0.065	2	1.62
9	1.46	0.29	0.079	2	1.83
12	1.52	0.38	0.097	2	2
15	1.57	0.47	0.11	6	4.075
18	1.65	0.52	0.13	10	7.59
20	1.7	0.64	0.141	12	10.64
21	1.73	0.68	0.147	14	12.1
22	1.75	0.72	0.153	16	13.92
23	1.76	0.75	0.158	17	15.75
24	1.77	0.79	0.164	18	17.64

TABLE IV

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

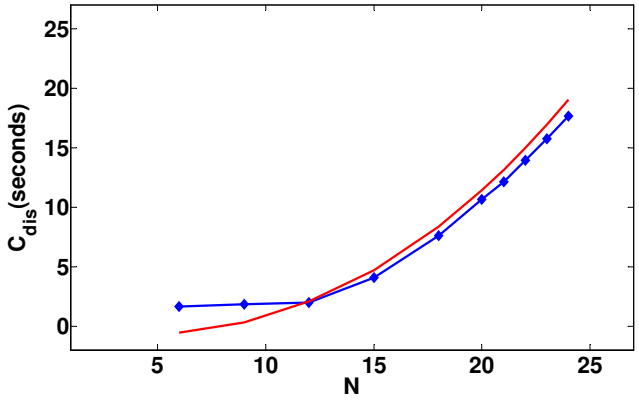

Fig. 4. $C_{dis}(N)$ versus the horizon length for $n = 42$. Blue dots are experimental data. Red curve includes the corresponding approximated data obtained by polynomial (16).

follows that there is a significant advantage in terms of the computation overhead in using the distributed algorithm for the irrigation network of interest to us. Specifically, when $nN > 400$ there is a significant reduction in computation overhead 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 overhead of a centralized optimization algorithm for solving the QP problem (2) with the computation overhead of the distributed optimization algorithm of [2]. It was assumed that both algorithms use the active set 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 overhead in using the distributed optimization algorithm of [2] for large-scale irrigation networks. Specifically, for the particular network of interest to us (the 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 overhead when

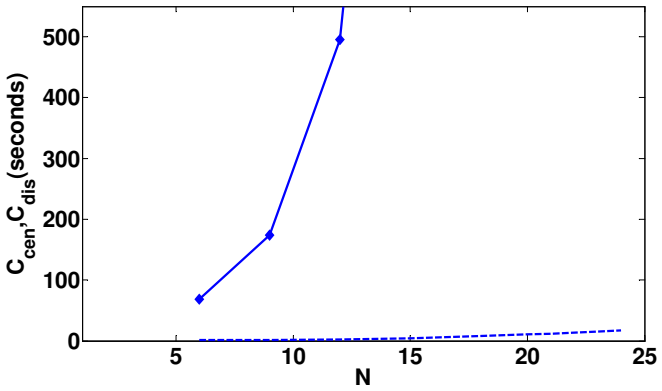

Fig. 5. $C_{cen}(N)$ and $C_{dis}(N)$ versus the horizon length N for $n = 42$. Solid curve indicates $C_{cen}(N)$ and dashed curve indicates $C_{dis}(N)$.

$nN > 400$ if the distributed optimization algorithm of [2] is used.

REFERENCES

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

[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] A. Farhadi, M. Cantoni, and P. M. Dower, Computation time analysis of a distributed optimization method applied to automated irrigation networks, *in Proceedings of 2013 IEEE Conference on Decision and Control*, Florence, Italy, December 10-13, 2013.

[4] 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.

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

[6] 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