Decentralized Multi-Agent Optimization via Dual Decomposition [⋆]

Håkan Terelius [*], **Ufuk Topcu** [**] **and Richard M. Murray** [**]

[*] *ACCESS Linnaeus Centre, Royal Institute of Technology, Stockholm, SE-100 44 Sweden (e-mail: hakante@kth.se)*
[**] *Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: {utopcu, murray}@cds.caltech.edu)*

Abstract: We study a distributed multi-agent optimization problem of minimizing the sum of convex objective functions. A new decentralized optimization algorithm is introduced, based on dual decomposition, together with the subgradient method for finding the optimal solution. The iterative algorithm is implemented on a multi-hop network and is designed to handle communication delays. The convergence of the algorithm is proved for communication networks with bounded delays. An explicit bound, which depends on the communication delays, on the convergence rate is given. A numerical comparison with a decentralized primal algorithm shows that the dual algorithm converges faster, with less communication.

Keywords: Distributed multi-agent systems, Optimization, Dual decomposition

1. INTRODUCTION

The ability to compute an optimal distributed decision has gained a lot of attention in recent research, e.g., Tsitsiklis (1984); Lynch (1996); Nedić and Ozdaglar (2007a,b); Johansson (2008). Distributed optimization problems appear in a broad range of practical applications, such as when a set of network users are competing for a shared resource, with different individual objectives. These problems are commonly described as a utility maximization problem, where each user has its own utility function. The goal of the network is to assign the resources such that the total utility is maximized. The users are referred to as *agents*, and the interconnected network of agents is called a *multi-agent system*. The multi-agent system consists of agents making local decisions, while trying to coordinate the overall objective with the other agents in the system.

Another application that has received a lot of attention is distributed cooperative control of unmanned vehicles, and especially, formation control of such vehicles, e.g., Olfati-Saber and Murray (2002, 2004); Fax and Murray (2004); Blondel et al. (2005). For example, using several micro-satellites has the potential for greater functionality and reliability than an individual satellite. Also, formation control of a group of unmanned aircrafts or underwater vehicles can severely increase the efficiency in surveillance and search-and-rescue operations. Common to all these examples is that the optimal control algorithms should be completely distributed, relying on only local information.

There are two approaches to decompose the underlying optimization problem; primal and dual decomposition. Nedić and Ozdaglar (2007a) has analyzed a decentralized optimization algorithm based on primal decomposition. In this paper, we propose and analyze a decentralized optimization algorithm based on the dual decomposition technique.

In Section 2, we describe the optimization problem and the corresponding dual optimization problem. In Section 3, we develop a novel decentralized algorithm based on the dual optimization problem, and in Section 4, we analyze the convergence rate of the algorithm. In Section 5, we study the communication requirements for the algorithm, and finally, in Section 6, we evaluate the algorithm with numerical simulations. An extended version of this work is available as Terelius (2010).

2. PROBLEM FORMULATION

We consider an optimization problem, where a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (Fig. 1) of N interconnected agents are minimizing the sum of their individual convex (potentially non-smooth) objective functions $f^i : \mathbb{R}^n \to \mathbb{R}$.

$$f^* := \min_{x \in X} \sum_{i=1}^{N} f^i(x), \tag{1}$$

where $X \subseteq \mathbb{R}^n$ is non-empty and convex. Each individual objective function f^i is assumed to be known only by agent i, thus the agents $\mathcal{V} = \{1, \ldots, N\}$ need to coordinate their decisions through a limited set of communication links $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The goal is to solve the minimization problem in a decentralized fashion, where the agents cooperatively find the optimal solution without a central coordinator. We propose a procedure based on dual decomposition. As a prelude to the proposed procedure, let us discuss the dual decomposition.

To this end, let us introduce a local estimate $x^i \in X$ of the common decision variable $x \in X$ for each agent $i \in \mathcal{V}$

[⋆] We thank Karl-Henrik Johansson and Henrik Sandberg for their support and helpful discussions. This work was partially supported by the Swedish Research Council, the Knut and Alice Wallenberg Foundation and the Boeing Corporation.

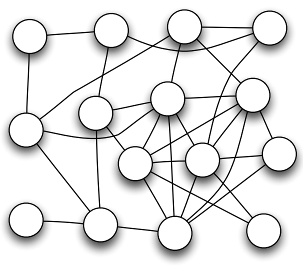

Fig. 1. A multi-agent system $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the agents $\mathcal{V}$ are pictured as vertices, and the communication links $\mathcal{E}$ between the agents are pictured as edges.

in the network. The optimization problem in (1) can then be written as

$$\begin{aligned} &\min_{x^1,\dots,x^N \in X} \sum_{i=1}^{N} f^i(x^i), \\ &\text{subject to } x^1 = x^2 = \cdots = x^N. \end{aligned}$$

Notice that the coupling between the objective functions are moved from the decision variable to the consistency constraints. Next, let us partition the decision variable $x \in X$ into N parts such that each part can be associated with a unique agent,

$$x = \left[x_1^T,\ x_2^T, \dots, x_N^T\right]^T,$$

where $x_i \in \mathbb{R}^{n_i}$ for $i \in \mathcal{V}$, $n_i \geq 0$ and $\sum_{i=1}^N n_i = n$. This partitioning can be arbitrary, however, x_i is associated with agent i, and in many applications there is a natural partitioning, where x_i represents the internal state of agent i. Thus, x_j^i would denote agent i's estimate of agent j's internal state. We now introduce the dual variables λ_{ij}, $i, j \in \mathcal{V}$, where λ_{ij} is associated with the constraint $x_i^i = x_i^j$. For notational simplicity, let us also introduce λ as

$$\lambda := \left[\lambda_{11}^T, \dots, \lambda_{1N}^T,\ \lambda_{21}^T, \dots, \lambda_{2N}^T, \dots, \lambda_{NN}^T\right]^T.$$

The Lagrangian L is defined as

$$\begin{aligned} L(x^1, x^2, \dots, x^N, \lambda) :=& \sum_{i=1}^{N} f^i(x_1^i, x_2^i, \dots, x_N^i) \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda_{ij}^T \left(x_i^i - x_i^j\right), \end{aligned}$$

and the corresponding Lagrange dual function q is the infimum with respect to the primal variables $x^1, \dots, x^N$,

$$q(\lambda) := \inf_{x^1, x^2, \dots, x^N \in X} L(x^1, x^2, \dots, x^N, \lambda).$$

Remark 1. Notice that L is independent of the dual variables λ_{ii}, since λ_{ii} accounts for the difference $x_i^i - x_i^i = 0$. Thus, λ_{ii} could be removed from the expression, but for notational simplicity, we instead define $\lambda_{ii} = 0$.

Let us now rewrite the Lagrange dual function by introducing the subproblems $\phi^i(\lambda)$ as

$$\phi^i(\lambda) := \inf_{x^i \in X} f^i(x_1^i, x_2^i, \dots, x_N^i) + \sum_{j=1}^{N} \lambda_{ij}^T x_i^i - \sum_{j=1}^{N} \lambda_{ji}^T x_j^i. \tag{2}$$

Notice that the subproblem ϕ^i only depends on the dual variable λ, and is the infimum over the local variable x^i. Thus, agent i is able to compute $\phi^i(\lambda)$ locally, independent of all other agents. Further, the Lagrange dual function is the sum of all the subproblems,

$$q(\lambda) = \sum_{i=1}^{N} \phi^i(\lambda). \tag{3}$$

Finally, the Lagrange dual problem is the maximization of the Lagrange dual function,

$$q^* := \max_{\lambda} q(\lambda) = \max_{\lambda} \sum_{i=1}^{N} \phi^i(\lambda). \tag{4}$$

Slater's condition (see Boyd and Vandenberghe (2004)) guarantees that $q^* = f^*$ under the convexity assumptions, and hence, we can solve the Lagrange dual problem instead of the original optimization problem.

3. DUAL OPTIMIZATION ALGORITHM

We solve the Lagrange dual problem (4) with the subgradient method. The subgradient method is a generalization of the gradient descent method to non-differentiable functions, using the iterations

$$\lambda(t+1) = \lambda(t) + \alpha_t g(t), \tag{5}$$

where α_t is the step size used at time t, and $g(t)$ is a subgradient to q at $\lambda(t)$, i.e.,

$$q(\bar{\lambda}) \leq q(\lambda(t)) + g(t)^T(\bar{\lambda} - \lambda(t)), \tag{6}$$

holds for all $\bar{\lambda}$.

Let us consider the subgradient update (5) for a single component λ_{ij} of the dual variable. The subgradient to q with respect to λ_{ij} is $x_i^i - x_i^j$, thus the subgradient update can be written as

$$\lambda_{ij}(t+1) = \lambda_{ij}(t) + \alpha_t \left(x_i^i(t) - x_i^j(t)\right), \tag{7}$$

where $x^i(t)$ is given by the solution to the subproblem ϕ^i, evaluated at $\lambda(t)$.

Since the subproblems can be solved independently of each other, the remaining part of the algorithm is to update the dual variables, λ, in a decentralized manner. We assume that all agents perform their computations and communications synchronously, at the discrete times $t = 0, 1, \dots$. Further, we assume that only neighbors can communicate with each other during one time step.

Let us return to the subgradient update rule (7), and notice that both agent i's estimate x_i^i and agent j's estimate x_i^j are necessary to update the dual variable λ_{ij}. Thus, the solution is that agent i sends its estimate to agent j, who then is able to calculate the dual variable. Because the two agents might not be neighbors, we introduce the *time-delay* δ_{ij} that measures the multi-hop delay from agent i to agent j. Further, let $d_{ij} := \delta_{ij} + \delta_{ji}$ denote the *round-trip time* between agent i and agent j, i.e., the time it takes agent j to respond to agent i.

The update rule (7) is now modified to account for the communication delays. Let $\lambda_{ij}(t)$ denote the *commonly known dual variable* that is known to both agent i and agent j at time t, which they also use to solve their respective subproblem ϕ^i and ϕ^j. The solutions to the subproblems, given $\lambda(t)$, is denoted by $x^i(t)$ and $x^j(t)$.

Agent i transmits the part $x_i^i(t)$ to agent j, who receives it at time $t+\delta_{ij}$. Agent j is then able to update the dual variable λ_{ij}, which it transmits back to agent i. The total time it takes to update the commonly known dual variable is equal to the round trip time d_{ij} between agent i and agent j. The update can then formally be expressed as

$$\lambda_{ij}(t+d_{ij}+1) = \lambda_{ij}(t+d_{ij}) + \alpha_t \left(x_i^i(t) - x_i^j(t) \right). \quad (8)$$

Remark 2. The commonly known dual variable does not necessarily satisfy the subgradient condition (6), and this is the main problem that we analyze in the next section.

Remark 3. Different parts of the dual variable λ can be updated with different delays, but each part is updated exactly once at each iteration. Also, each agent solves its subproblem exactly once at each iteration.

Before we move into the analysis of the algorithm, we state four assumptions about the problem instance. Assumptions 1-3 are commonly used for the subgradient method and Assumption 4 is due to the time-delays we introduced.

Assumption 1. (Existence of Maximizer). There exists at least one finite maximizer of q, denoted by λ^*. Let Λ_{opt} denote the set of maximizers to q.

Assumption 2. (Bounded Subgradients). For every subgradient $g(t)$ to q at $\lambda(t)$, $g(t)$ is uniformly bounded by G,

$$||g(t)||_2 \le G \quad \forall t.$$

Remark 4. Assumption 2 automatically holds if the set X is compact.

Assumption 3. (Bounded Initial Distance). The distance from the initial point, $\lambda(0)$, to the optimal set is bounded by R,

$$d\left(\lambda(0), \Lambda_{opt}\right) \le R.$$

In order for the algorithm to work, it is necessary to assume that the network is strongly connected. We impose this by assuming that the time-delays are bounded, which also implies that no packages are lost in the network.

Assumption 4. (Bounded Time-Delays). There exists an upper bound D on the round-trip time,

$$d_{ij} = \delta_{ij} + \delta_{ji} \le D \quad \forall i,j \in \mathcal{V}.$$

4. CONVERGENCE ANALYSIS

The analysis focuses on the evolution of the dual variables $\lambda(t)$, and recall that $g(t)$ is a subgradient to the Lagrange dual function, q, evaluated at $\lambda(t)$. Let us define the shifted vector of dual variables $\bar\lambda(t)$ by

$$\bar\lambda_{ij}(t) := \lambda_{ij}(t+d_{ij}),$$

thus, the entire vector $\bar\lambda(t)$ is

$$\bar\lambda(t) = \begin{bmatrix} \lambda_{11}(t+d_{11}) \\ \vdots \\ \lambda_{ij}(t+d_{ij}) \\ \vdots \\ \lambda_{NN}(t+d_{NN}) \end{bmatrix}.$$

The update rule (8) can be compactly written as

$$\bar\lambda(t+1) = \bar\lambda(t) + \alpha_t g(t). \quad (9)$$

Remark 5. Notice the similarity between this expression and the ordinary subgradient method iteration (5), however, remember that $g(t)$ is a subgradient at $\lambda(t)$ and not at $\bar\lambda(t)$.

Let us express the entire dual vector $\lambda(t)$ in terms of $\bar\lambda(t)$ with the following vector projection P_d. Let $P_d(x)$ be defined by

$$[P_d(x)]_{ij} := \begin{cases} x_{ij} & \text{if } d_{ij} = d; \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $P_d(x)$ selects those components of x whose round-trip time is equal to d. Notice two important properties of P under Assumption 4. First, for any vector x

$$\sum_{d=0}^{D} P_d(x) = x. \quad (10)$$

Second, we can express $\lambda(t)$ from $\bar\lambda(t)$ as

$$\sum_{d=0}^{D} P_d(\bar\lambda(t-d)) = \lambda(t). \quad (11)$$

Lemma 1. Under Assumption 3,

$$d\left(\bar\lambda(0), \Lambda_{opt}\right) \le R.$$

Proof. By the definition of $\bar\lambda$, we have

$$\bar\lambda_{ij}(0) = \lambda_{ij}(d_{ij}).$$

Notice that it takes d_{ij} time steps until agent i receives its first update on the dual variable λ_{ij}, hence $\lambda_{ij}(0) = \cdots = \lambda_{ij}(d_{ij})$ and $\bar\lambda(0) = \lambda(0)$. The lemma now follows from Assumption 3. □

Lemma 2. Under Assumption 2,

$$\left|\left|\bar\lambda(t+1) - \bar\lambda(t)\right|\right|_2 \le \alpha_t G.$$

Proof. From the update rule (9), we have

$$\bar\lambda(t+1) - \bar\lambda(t) = \alpha_t g(t).$$

Thus, by Assumption 2,

$$\left|\left|\bar\lambda(t+1) - \bar\lambda(t)\right|\right|_2 = \alpha_t ||g(t)||_2 \le \alpha_t G. \qquad □$$

In the next lemma we give an upper bound on the difference between $\bar\lambda(t)$ and $\lambda(t)$.

Lemma 3. Under Assumption 2 and 4,

$$\left|\left|\bar\lambda(t) - \lambda(t)\right|\right|_2 \le G \sum_{d=t-D}^{t-1} \alpha_d.$$

Proof. Recall that $\bar\lambda(t) - \lambda(t)$ can be written as

$$\begin{aligned} \bar\lambda(t) - \lambda(t) &= \bar\lambda(t) - \sum_{d=0}^{D} P_d\left(\bar\lambda(t-d)\right) \\ &= \sum_{d=1}^{D} P_d\left(\bar\lambda(t) - \bar\lambda(t-d)\right). \end{aligned}$$

By replacing the term $\bar\lambda(t) - \bar\lambda(t-d)$ with a telescoping sum, we have

$$\begin{aligned} \bar\lambda(t) - \lambda(t) &= \sum_{d=1}^{D} P_d\left(\sum_{i=0}^{d-1}\left(\bar\lambda(t-i) - \bar\lambda(t-i-1)\right)\right) \\ &= \sum_{d=1}^{D}\sum_{i=0}^{d-1} P_d\left(\bar\lambda(t-i) - \bar\lambda(t-i-1)\right), \end{aligned}$$

since P_d is a linear function. Now, by changing the order of summation in this double sum,

$$\bar{\lambda}(t) - \lambda(t) = \sum_{i=0}^{D-1} \sum_{d=i+1}^{D} P_d \left(\bar{\lambda}(t-i) - \bar{\lambda}(t-i-1)\right).$$

By the triangle inequality,

$$\left|\left|\bar{\lambda}(t) - \lambda(t)\right|\right|_2 \leq \sum_{i=0}^{D-1} \left|\left| \sum_{d=i+1}^{D} P_d \left(\bar{\lambda}(t-i) - \bar{\lambda}(t-i-1)\right)\right|\right|_2$$

Further, since the projections P_d are disjoint for different d,

$$\begin{aligned} &\left|\left| \sum_{d=i+1}^{D} P_d \left(\bar{\lambda}(t-i) - \bar{\lambda}(t-i-1)\right)\right|\right|_2 \\ &\leq \left|\left| \sum_{d=0}^{D} P_d \left(\bar{\lambda}(t-i) - \bar{\lambda}(t-i-1)\right)\right|\right|_2 \\ &= \left|\left|\bar{\lambda}(t-i) - \bar{\lambda}(t-i-1)\right|\right|_2. \end{aligned}$$

Using Lemma 2 yields

$$\left|\left|\bar{\lambda}(t-i) - \bar{\lambda}(t-i-1)\right|\right|_2 \leq G\alpha_{t-i-1}.$$

Assembling everything together gives

$$\left|\left|\bar{\lambda}(t) - \lambda(t)\right|\right|_2 \leq \sum_{i=0}^{D-1} G\alpha_{t-i-1} = G \sum_{d=t-D}^{t-1} \alpha_d. \qquad \square$$

Remark 6. If $t < D$, then $\lambda(t)$ should be written as $\lambda(t) = \sum_{d=0}^{t} P_d(\bar{\lambda}(t-d))$, and the bound becomes

$$\left|\left|\bar{\lambda}(t) - \lambda(t)\right|\right|_2 \leq \sum_{i=0}^{t-1} G\alpha_{t-i-1} = G \sum_{d=0}^{t-1} \alpha_d.$$

Instead of treating this case separately, it is easier to define $\alpha_t = 0$ if $t < 0$.

Lemma 4. Under Assumption 2 and 4,

$$g(t)^T \left(\bar{\lambda}(t) - \lambda^*\right) \leq G^2 \sum_{d=t-D}^{t-1} \alpha_d + q(\lambda(t)) - q^*. \qquad (12)$$

Proof. Adding $g(t)^T(\lambda(t) - \lambda(t)) = 0$ to the left-hand side of equation (12) yields

$$g(t)^T \left(\bar{\lambda}(t) - \lambda^*\right) = g(t)^T \left(\bar{\lambda}(t) - \lambda(t)\right) + g(t)^T \left(\lambda(t) - \lambda^*\right)$$

By Cauchy-Schwarz inequality,

$$g(t)^T \left(\bar{\lambda}(t) - \lambda(t)\right) \leq ||g(t)||_2 \cdot \left|\left|\bar{\lambda}(t) - \lambda(t)\right|\right|_2.$$

Since $g(t)$ is a subgradient at $\lambda(t)$, the subgradient definition (6) implies that

$$g(t)^T \left(\lambda(t) - \lambda^*\right) \leq q(\lambda(t)) - q^*.$$

Thus,

$$g(t)^T \left(\bar{\lambda}(t) - \lambda^*\right) \leq ||g(t)||_2 \cdot \left|\left|\bar{\lambda}(t) - \lambda(t)\right|\right|_2 + q(\lambda(t)) - q^*.$$

Finally, using Lemma 3 and Assumption 2, we get

$$g(t)^T \left(\bar{\lambda}(t) - \lambda^*\right) \leq G^2 \sum_{d=t-D}^{t-1} \alpha_d + q(\lambda(t)) - q^*. \qquad \square$$

The function value $q(\lambda(t))$ does not have to be monotonically increasing. Therefore, the algorithm is evaluated as the maximum value over a period of T iterations. Let us now state and prove the main convergence theorem for the dual optimization algorithm.

Theorem 5. Let Assumptions 1,2, 3 and 4 hold. Then, the maximum value satisfies the following bound,

$$\max_{t=0,\dots,T} q(\lambda(t)) \geq q^* - \frac{R^2 + G^2 \sum_{t=0}^{T} \left(\alpha_t^2 + 2\alpha_t \sum_{d=t-D}^{t-1} \alpha_d\right)}{2\sum_{t=0}^{T} \alpha_t}. \qquad (13)$$

Proof. Let λ^* be any optimal point to q, and let $\bar{\lambda}(t)$ be given by the iterations (9). Consider the following relation

$$\begin{aligned} 0 &\leq \left|\left|\bar{\lambda}(T+1) - \lambda^*\right|\right|_2^2 = \left|\left|\bar{\lambda}(T) + \alpha_T g(T) - \lambda^*\right|\right|_2^2 \\ &= \left|\left|\bar{\lambda}(T) - \lambda^*\right|\right|_2^2 + 2\alpha_T g(T)^T \left(\bar{\lambda}(T) - \lambda^*\right) + \alpha_T^2 ||g(T)||_2^2. \end{aligned}$$

This is a recursive equation in $\left|\left|\bar{\lambda}(t) - \lambda^*\right|\right|_2^2$, and can be expanded until $t = 0$, yielding

$$0 \leq \left|\left|\bar{\lambda}(0) - \lambda^*\right|\right|_2^2 + \sum_{t=0}^{T} 2\alpha_t g(t)^T \left(\bar{\lambda}(t) - \lambda^*\right) + \sum_{t=0}^{T} \alpha_t^2 ||g(t)||_2^2.$$

Using equation (12) from Lemma 4 gives us

$$\begin{aligned} 0 &\leq \left|\left|\bar{\lambda}(0) - \lambda^*\right|\right|_2^2 + \sum_{t=0}^{T} 2\alpha_t \left(G^2 \sum_{d=t-D}^{t-1} \alpha_d + q(\lambda(t)) - q^*\right) \\ &\qquad + \sum_{t=0}^{T} \alpha_t^2 ||g(t)||_2^2 \\ &\leq \left|\left|\bar{\lambda}(0) - \lambda^*\right|\right|_2^2 + \sum_{t=0}^{T} 2\alpha_t G^2 \sum_{d=t-D}^{t-1} \alpha_d \\ &\qquad + \left(\max_{t=0,\dots,T} q(\lambda(t)) - q^*\right) \sum_{t=0}^{T} 2\alpha_t + \sum_{t=0}^{T} \alpha_t^2 ||g(t)||_2^2. \end{aligned}$$

Thus,

$$\begin{aligned} \max_{t=0,\dots,T} q(\lambda(t)) &\geq q^* - \frac{\left|\left|\bar{\lambda}(0) - \lambda^*\right|\right|_2^2}{\sum_{t=0}^{T} 2\alpha_t} \\ &\quad + \frac{\sum_{t=0}^{T} 2\alpha_t G^2 \sum_{d=t-D}^{t-1} \alpha_d + \sum_{t=0}^{T} \alpha_t^2 ||g(t)||_2^2}{\sum_{t=0}^{T} 2\alpha_t}. \end{aligned}$$

Notice that λ^* is an arbitrary optimal point, thus $\left|\left|\bar{\lambda}(0) - \lambda^*\right|\right|_2$ can be replaced with $d\left(\bar{\lambda}(0), \Lambda_{opt}\right)$. Finally, using Lemma 1 and Assumption 2 gives us

$$\begin{aligned} &\max_{t=0,\dots,T} q(\lambda(t)) \\ &\geq q^* - \frac{R^2 + \sum_{t=0}^{T} 2\alpha_t G^2 \sum_{d=t-D}^{t-1} \alpha_d + \sum_{t=0}^{T} \alpha_t^2 G^2}{\sum_{t=0}^{T} 2\alpha_t} \\ &= q^* - \frac{R^2 + G^2 \sum_{t=0}^{T} \left(\alpha_t^2 + 2\alpha_t \sum_{d=t-D}^{t-1} \alpha_d\right)}{2\sum_{t=0}^{T} \alpha_t}. \qquad \square \end{aligned}$$

Corollary 6. For a constant step size $\alpha_t = \alpha$, the convergence result in Theorem 5 becomes

$$\max_{t=0,\dots,T-1} q(\lambda(t)) \geq q^* - \frac{R^2 + G^2\alpha^2 T(1+2D)}{2\alpha T}. \qquad (14)$$

By letting $D = 0$ in Corollary 6, we recognize the convergence result for the ordinary subgradient method, without any delays. The conclusion is that the delays do not affect the convergence speed, but instead cause the algorithm to converge to a larger neighborhood around the optimal solution. The intuition behind the result is that the ordinary subgradient method overshoots the optimal point with half the step length, while using information delayed D steps can cause the algorithm to continue past the optimal point for an additional D steps.

Corollary 7. Consider also a square summable, but not summable step size $\alpha_t \geq \alpha_{t+1} \geq 0 \quad t = 0, 1, \ldots$, such that $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ and $\sum_{t=0}^{\infty} \alpha_t = \infty$. Then, the convergence result in Theorem 5 becomes

$$\max_{t=0,\ldots,T} q(\lambda(t)) \geq q^* - \frac{R^2 + G^2(1+2D)\sum_{i=0}^{t} \alpha_{i-D}^2}{2\sum_{i=0}^{t} \alpha_i} \;\rightarrow\; q^*$$

as $t \rightarrow \infty$.

5. COMMUNICATION COST

In distributed systems, such as wireless sensor networks, the limiting resource is often the communication. It is therefore important to analyze the convergence speed in relation to the communication. For simplicity, we define the communication cost of a transmission as the number of real valued scalars transfered. There are two contributions to the communication cost: distributing the primal variables $x_i^i(t)$, and accumulating the dual variables λ_{ji}. Let us analyze these contributions separately, per iteration of the algorithm.

Recall that only the part $x_i^i(t)$ of agent i's estimate $x^i(t)$ has to be distributed to all other agents in the network, in order to update the dual variables. A reasonable assumption is that an agent only receives the estimate $x_i^i(t)$ once. Since there are N agents in the network, each part is transfered over exactly $N-1$ edges. Summing over all parts yield the communication cost

$$\sum_{i=1}^{N} (N-1)\dim(x_i^i) = (N-1)\dim(x) = (N-1)n.$$

In contrast to the primal variables, there are many more dual variables, but each dual variable λ_{ij} needs only to be transfered from agent j to agent i. With the assumption that each agent only receives an estimate once, we similarly have the communication cost

$$\sum_{i=1}^{N}\sum_{j=1}^{N} (N-1)\dim(\lambda_{ij}) = (N-1)\dim(\lambda).$$

To improve upon this, notice that the local objective function $\phi^i(\lambda)$ only depends on the dual variables λ_{ji}, $j \in \mathcal{V}$, and the sum $\sum_{j=1}^{N} \lambda_{ij}$, not the individual values λ_{ij}, $j \in \mathcal{V}$. Further, since the dual variables λ_{ji} are calculated at agent i, it only needs to receive the sum of the dual variables $\sum_{j=1}^{N} \lambda_{ij}$.

Thus, an agent k can sum all the received dual variables λ_{ij} directed towards agent i, add its own dual variable λ_{ik} to this sum, and then transfer the entire sum, instead of the individual values. All dual variables λ_{ij} directed towards agent i are then together only transfered over $N-1$ edges, and the total cost of transporting all dual variables is

$$\begin{aligned}\sum_{i=1}^{N} (N-1)\dim(\lambda_{ij}) &= \sum_{i=1}^{N} (N-1)\dim(x_i) \\ &= (N-1)n.\end{aligned}$$

Adding the communication cost for the primal and dual variables leads to the following proposition.

Proposition 8. The total communication cost for the dual optimization algorithm is

$$2(N-1)n. \tag{15}$$

Remark 7. Notice that the communication cost is independent of the network topology, however, we have neglected the communication cost for routing overhead.

The last communication consideration that we want to emphasize is related to the problem structure. Recall that each local objective function can be written as $f^i(x_1, x_2, \ldots, x_N)$, where all objective functions f^i are assumed to depend on all of the decision variables $x_1, x_2, \ldots, x_N$. In practice, many interesting distributed optimization problems have a sparse structure in the dependencies. Assume that f^i does not depend upon x_j, i.e., $f^i(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$. Then the variable x_j^i does not exist, and hence, there is no need for the dual variable λ_{ji}, accounting for the difference between x_j^j and x_j^i. The implications for the communication is that the primal variable x_j^j does not have to be sent to agent i, and the dual variable λ_{ji} does not have to be sent back to agent j, resulting in an even lower communication cost.

6. NUMERICAL RESULTS

6.1 A Primal Distributed Optimization Algorithm

We compare the dual optimization algorithm against a decentralized optimization algorithm, analyzed by Nedić and Ozdaglar (2007b). We refer to this algorithm as the *primal algorithm.* The primal algorithm is also an iterative algorithm, where each agent has a local estimate $x^i(t)$ of the decision variable, but the update consists of an average consensus update together with a subgradient step for the local objective function, i.e.,

$$x^i(t+1) = \sum_{j=1}^{N} W_{j,i}(t)x^j(t) - \alpha_t^i g^i(t), \tag{16}$$

where W is an average consensus matrix, and $g^i(t)$ is a subgradient to f^i at $x^i(t)$.

6.2 Evaluating the Convergence Rate

In order to compare the convergence rate, we need a measure of convergence. For both algorithms, each agent has an estimate of the entire decision variable $x^i(t)$, but we are seeking a solution where all of these estimates are equal. Therefore, we evaluate the algorithms on the average of the agents estimates, let

$$x_{\text{avg}}(t) := \frac{1}{N}\sum_{i=1}^{N} x^i(t).$$

The function value is evaluated on the average state estimate,

$$f(t) := \sum_{i=1}^{N} f^i(x_{\text{avg}}(t)).$$

Finally, since the function value does not need to be monotonically decreasing, we evaluate each algorithm as the minimum value attained until time t,

$$f_{\text{best}}(t) := \min_{i=0,\ldots,t} f(i).$$

For the dual optimization algorithm, we can also evaluate the Lagrange dual function on the commonly known dual variable $\lambda(t)$,

$$q(t) := \sum_{i=1}^{N} \phi^i(\lambda(t)).$$

Also, since the dual function does not need to be monotonically increasing, we evaluate the dual value as the maximum attained until time t,

$$q_{\text{best}}(t) := \max_{i=0,\ldots,t} q(i).$$

Both algorithms' convergence rate depends on their respective step size rule. We limit both algorithms to use a constant step size $\alpha_t = \alpha$, and then selects the optimal constant step size for each algorithm. Thus, for a given time interval $0, \ldots, T$, we use a step size α such that $f_{\text{best}}(T)$ is minimal.

6.3 Connected Graph

We start with a randomly chosen, strongly connected, network consisting of 15 agents (Fig. 1). The objective functions f^i are chosen as random convex quadratic functions, where the decision variable is constrained to a compact set X. The maximal delay between any two agents is 3, and hence, the upper bound on the round-trip time is $D = 6$. The step sizes are chosen such that the best function value $f_{\text{best}}(T)$ at time step $T = 1000$ is minimal, which results in the primal algorithm using the step size $\alpha = 0.00069$, and the dual algorithm using the step size $\alpha = 0.0568$.

The convergence rate is evaluated on the time interval $t = 0, \ldots, 2000$ (Fig. 2). Notice that the primal algorithm reaches a neighborhood around the optimal value at time step 1000, and does not significantly improve during the next 1000 steps. It is a tradeoff in step size; a smaller step size would yield a slower convergence, but to a smaller neighborhood, thus, giving a worse value at time step 1000, but a better solution at time step 2000. The dual algorithm, on the other hand, is close to the optimal value already after 1000 steps, and has essentially reached the optimal value after 2000 steps.

Finally, let us compare the communication cost for each algorithm. Notice that each agent in the primal algorithm transmits its local estimate to all available neighbors, hence the communication cost for the primal algorithm is $|\mathcal{E}| \dim(x)$, where the edges are counted with direction. The network in Fig. 1 has 62 edges, counted once in each direction, and hence, the communication cost is $62n$ for the primal, and $28n$ for the dual algorithm.

Thus, for this problem instance, the dual algorithm converges faster, to a smaller neighborhood around the optimal value, and with less communication than the primal algorithm.

6.4 Line Graph

Next, we consider a network with a line topology (Fig. 3). A line graph with N agents is a connected graph with diameter $N - 1$, and thus, has the highest round-trip time among all connected graphs of N agents. The round-trip time is bounded by $D = 26$. The optimization

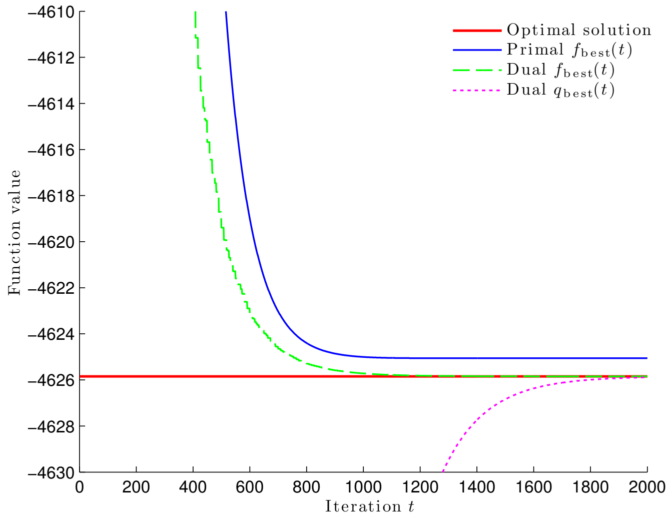

Fig. 2. Simulating the convergence rate on the interval $t = 0, \ldots, 2000$, with the optimal step sizes chosen at step $t = 1000$. The agents are communicating through a connected network (Fig. 1). The primal algorithm reaches a neighborhood around the optimal value, while the dual algorithm converges to the optimal value.

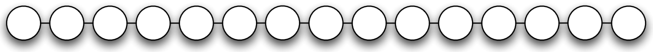

Fig. 3. The multi-agent system $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is communicating through a line topology. This topology has the maximal round-trip time among all networks with N agents.

problem is otherwise chosen as in the previous section. The optimal step sizes are chosen at time $T = 1000$, and the convergence rate is once again evaluated on the time interval $t = 0, \ldots, 2000$ (Fig. 4).

The results here are almost the opposite compared to the previous example. The primal algorithm converges to a lower value than the dual algorithm. The dual algorithm, on the other hand, reaches a neighborhood around the optimal value before step 1000, and does not improve after that. Notice the characteristic staircase appearance in the graph, it is caused by oscillations in the underlying function values $f(t)$, which are due to the delays in the network. Further, the number of directed edges is 28, thus the communication cost for both the primal and dual algorithm are equal to $28n$.

Thus, for this problem, the primal algorithm converges to a smaller neighborhood around the optimal value, and with the same communication cost as the dual algorithm.

6.5 Circular Graph

As a last example, we consider a network with circular topology (Fig. 5), which can be obtained from the line graph by adding only a single edge. This edge, however, causes the diameter of the graph to shrink from 14 to 7, and the upper bound on the round-trip time is $D = 12$ instead of 26. The optimization problem is again chosen as in the previous sections, and the convergence rate is evaluated on the interval $t = 0, \ldots, 2000$ (Fig. 6).

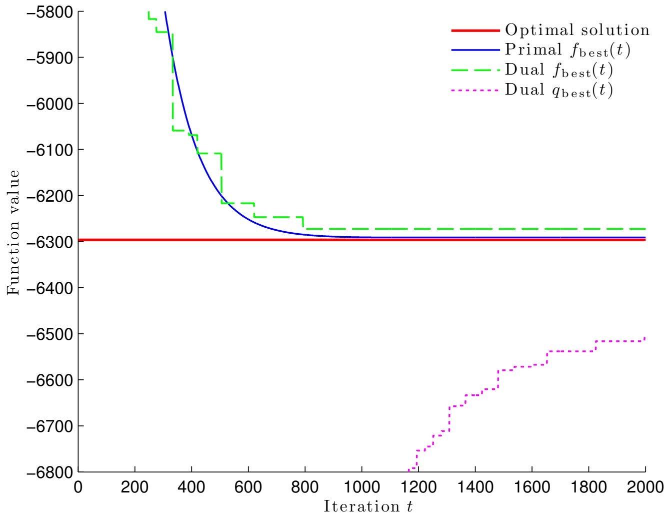

Fig. 4. Simulating the convergence rate on the interval $t = 0, \dots, 2000$, with the optimal step sizes chosen at step $t = 1000$. The agents are communicating through a line topology (Fig. 3). The primal algorithm attains a lower value than the dual algorithm. The staircase characteristics of the dual algorithm is typical for oscillations in the function value $f(t)$, caused by the delays.

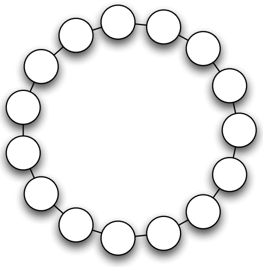

Fig. 5. The multi-agent system $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is communicating through a circular topology. It has one additional edge compared to the line topology, but the round-trip time is halved.

The results are similar to those of the first example, where the primal algorithm reaches a neighborhood around the optimal value before step 1000, while the dual algorithm converges to the optimal value. Also, the communication cost for the primal algorithm is $30n$ compared to the $28n$ for the dual algorithm.

7. CONCLUSIONS

Distributed optimization problems appear in a broad range of practical applications. In this paper, we have developed a novel decentralized optimization algorithm, based on the dual decomposition technique. Our analysis of the algorithm shows that it converges to the optimal solution when the communication delay is bounded. The numerical simulations show some promising results for our novel algorithm, both in terms of convergence speed and in decreasing the communication cost, especially for networks with a small diameter.

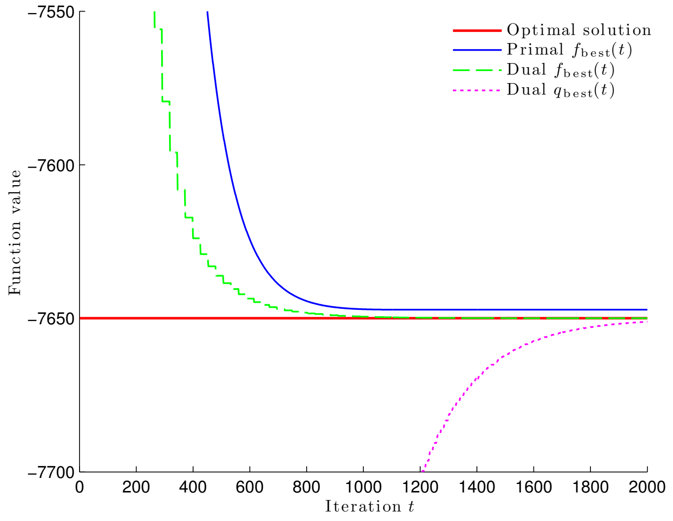

Fig. 6. Simulating the convergence rate on the interval $t = 0, \dots, 2000$, with the optimal step sizes chosen at step $t = 1000$. The agents are communicating through a circular topology (Fig. 5). The primal algorithm has reached a neighborhood around the optimal value, while the dual algorithm converges to the optimal value.

REFERENCES

Blondel, V.D., Hendrickx, J.M., Olshevsky, A., and Tsitsiklis, J.N. (2005). Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of Joint 44th IEEE Conference on Decision and Control and European Control Conference*, 2996–3000.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Fax, J.A. and Murray, R.M. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9), 1465–1476.

Johansson, B. (2008). *On Distributed Optimization in Networked Systems*. Ph.D. thesis, Royal Institute of Technology (KTH). TRITA-EE 2008:065.

Lynch, N.A. (1996). *Distributed algorithms*. Morgan Kaufmann.

Nedić, A. and Ozdaglar, A. (2007a). Distributed subgradient methods for multi-agent optimization. Technical Report 2575, MIT LIDS.

Nedić, A. and Ozdaglar, A. (2007b). On the rate of convergence of distributed subgradient methods for multi-agent optimization. In *Proceedings of 46th IEEE Conference on Decision and Control*, 4711–4716.

Olfati-Saber, R. and Murray, R.M. (2002). Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*. Barcelona, Spain.

Olfati-Saber, R. and Murray, R.M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), 1520–1533.

Terelius, H. (2010). *Distributed Multi-Agent Optimization via Dual Decomposition*. Master's thesis, Royal Institute of Technology, Stockholm. XR-EE-RT 2010:013.

Tsitsiklis, J.N. (1984). *Problems in Decentralized Decision Making and Computation*. Ph.D. thesis, Department of EECS, MIT.