# ROBUST DECENTRALIZED DYNAMIC OPTIMIZATION

*Wei Xu*[*] *Zhengqing Li* [*] *Qing Ling*[†]

[*] Department of Automation, University of Science and Technology of China
[†] School of Data and Computer Science, Sun Yat-Sen University

## ABSTRACT

This paper considers the problem of tracking a network-wide solution that dynamically minimizes the summation of time-varying local cost functions of agents, when some of the agents are malfunctioning. The malfunctioning agents broadcast faulty values to their neighbors, and lead the optimization process to a wrong direction. To mitigate the influence of the malfunctioning agents, we propose a total variation (TV) norm regularized formulation that drives the local variables of the regular agents to be close, while allows them to be different with the faulty values broadcast by the malfunctioning agents. We give a sufficient condition under which consensus of the regular agents is guaranteed, and bound the gap between the consensual solution and the optimal solution we pursue as if the malfunctioning agents do not exist. A fully decentralized subgradient algorithm is proposed to solve the TV norm regularized problem in a dynamic manner. At every time, every regular agent only needs one subgradient evaluation of its current local cost function, in addition to combining messages received from neighboring regular and malfunctioning agents. The tracking error is proved to be bounded, given that the variation of the optimal solution is bounded. Numerical experiments demonstrate the robust tracking performance of the proposed algorithm at presence of the malfunctioning agents.

***Index Terms*—** Decentralized networks, dynamic optimization, robust optimization, malfunctioning agents

## 1. INTRODUCTION

Consider an undirected network consisting of $n$ agents, which at time $k$ try to cooperatively solve a decentralized dynamic optimization problem

$$\min_{\tilde{x}^k} \sum_{i=1}^{n} f_i^k(\tilde{x}^k). \tag{1}$$

Here $f_i^k : \mathbb{R}^p \to \mathbb{R}$ is a convex and differentiable local cost function only available to agent $i$ at time $k$ and $\tilde{x}^k \in \mathbb{R}^p$ is the common optimization variable to all agents. At time $k$, every agent is allowed to exchange its current local iterate with network neighbors, followed by local computation so as to track the dynamic optimal solution. The purpose of this paper is to develop a robust decentralized dynamic optimization algorithm that solves (1) at presence of malfunctioning agents. By malfunctioning agents, we mean those who, instead of transmitting local iterates to neighbors, send wrong values (for example, faulty constants or random variables) due to failures of communication or computation units.

Decentralized dynamic optimization problems in the form of (1) are popular in multi-agent networks with time-varying tasks [1, 2, 3]. Examples include adaptive filtering and estimation in a wireless sensor network [4, 5, 6], target tracking using a group of robots [7, 8, 9], dynamic resource allocation over a communication network [10, 11, 12], voltage control of a power network [13, 14], to name a few. Existing algorithms to solve (1) are (sub)gradient methods [6, 13], alternating direction method of multipliers [1, 12], as well as gradient, Newton, and interior point methods based on the prediction-correction scheme [2, 3].

Nevertheless, most of the existing works assume that the agents faithfully follow prescribed optimization protocols: accessing dynamic local cost functions, exchanging local iterates, and performing local computations. This assumption does not always hold true since some of the agents might be unreliable in practice. Generally speaking, there are two kinds of unreliable agents:

1. *Malicious agents.* Agents send malicious information to their neighbors so as to deliberately guide the optimization process to a wrong direction that they expect to reach.
2. *Malfunctioning agents.* Agents send faulty values to their neighbors, not deliberately but due to failures of communication or computation units.

This paper focuses on handling malfunctioning agents in decentralized dynamic optimization. For works on mitigating the impact of malicious agents in adversarial environments, readers are referred to recent papers [15, 16, 17].

The impact of malfunctioning agents has been analyzed in the context of average consensus over a social network [18, 19, 20]. It is shown that the malfunctioning agents shall bias the network opinions from the consensual state of the regular agents [18], and the locations of the malfunctioning agents are critical to evolution of the network opinions [19]. Decentralized detection and localization methods are proposed in [20] to identify the malfunctioning agents. To the best of our knowledge, there is no existing work that considers the influence of the malfunctioning agents on the tracking performance of decentralized dynamic optimization.

Our work is tightly related to [21], whose goal is decentralized *static* optimization at presence of the malfunctioning agents. Different from the *dynamic* case studied in this paper, [21] assumes that the local cost functions $f_i^k$ are invariant across time $k$. To handle the faulty values broadcast by the malfunctioning agents, the total variation (TV) norm of the vector that stacks all the local variables is penalized. Through minimizing the summation of the local cost functions and the TV norm, most local variables (from the regular agents) are able to reach consensus and those outliers (from the malfunctioning agents) are tolerated. A subgradient method is proposed to solve this robust decentralized *static* optimization problem. Our work also adopts the TV norm penalty to handle the malfunctioning agents and a subgradient algorithm as the optimization tool, but extends their applications to the *dynamic* regime. We give a sufficient condition under which consensus of the regular agents is guaranteed, and also give an upper bound on the tracking error of the regular agents. These results provide theoretical guarantees to the tracking performance of the subgradient method at presence of the malfunctioning agents.

Another related work is [22], which considers decentralized *stochastic* optimization. Instead of tracking a dynamic optimal solution, [22] minimizes the summation of the local cost functions $f_i^k$ across all nodes $i$ and all times $k$. Therefore, the local iterates are expected to reach a steady-state consensual solution, given that the stochastic noise of the local cost functions is bounded. To allow for data heterogeneity across the network, [22] introduces proximity constraints such that neighboring local variables are close enough, but not necessarily consensual. Though not explicitly claimed in [22], this approach is also able to alleviate the influence of the malfunctioning agents. A saddle point method is proposed to solve this constrained stochastic optimization problem. Our work is different from [22] in terms of problem setting (dynamic versus stochastic), mathematical formulation (TV norm penalty versus proximity constraints), and algorithm design (subgradient versus saddle point).

The main contributions of this paper are as follows.

1. We formulate a TV norm regularized problem, which is robust to presence of the malfunctioning agents (Section 2). We give a sufficient condition under which consensus of the regular agents is guaranteed, and bound the gap between the consensual solution and the optimal solution we pursue as if the malfunctioning agents do not exist (Section 3, Theorem 1).

2. We propose a fully decentralized subgradient algorithm to solve the TV norm regularized problem in a dynamic manner. At every time, every regular agent only needs one subgradient evaluation, in addition to combining messages from neighboring regular and malfunctioning agents (Section 2). We prove that the tracking error is bounded, given that the variation of the optimal solution is bounded (Section 3, Theorem 2).

3. We provide extensive numerical experiments, demonstrating the robust tracking performance of the proposed algorithm at presence of the malfunctioning agents (Section 4).

## 2. FORMULATION AND ALGORITHM

Let us consider a connected undirected network of $n$ agents $\mathcal{V} = \{1, \cdots, n\}$ with $n = |\mathcal{V}|$, and a set of edges $\mathcal{A}$. If an edge $(i, j) \in \mathcal{A}$, then agents $i$ and $j$ are neighbors, and can communicate with each other. We denote the set of agent $i$'s neighbors as $\mathcal{N}_i$. The agents aim at solving the decentralized dynamic optimization problem in the from of (1). We assume that the network is synchronized, and at time $k$ every agent $i$ strictly conforms to the following protocol:

*Step 1.* Accessing local cost function $f_i^k$.

*Step 2.* Computing local iterate $x_i^k \in \mathbb{R}^p$.

*Step 3.* Broadcasting local iterate $x_i^k$ to neighbors $j \in \mathcal{N}_i$.

However, some of the agents in the network are malfunctioning, meaning that they broadcast faulty values other than local iterates. To be specific, denote $\mathcal{M}$ as the set of malfunctioning agents and $\mathcal{R} := \mathcal{V} \backslash \mathcal{M}$ as the set of regular agents. Define $r := |\mathcal{R}|$ and $m := |\mathcal{M}|$. The subset of edges connecting the regular agents in $\mathcal{V}$ is denoted by $\mathcal{E} \subseteq \mathcal{A}$. At time $k$, malfunctioning agent $i \in \mathcal{M}$ broadcasts a variable $z_i^k \in \mathbb{R}^p$, instead of $x_i^k$, to its neighbors $j \in \mathcal{N}_i$. The faulty value may come from failure of the computation unit, or breakdown of the communication unit. Different from [18,19,20,21] that assume the faulty values are constant across time $k$, we also allow that they are time-varying (for example, random variables or values generated from certain functions of time). Although identifying the malfunctioning agents is possible in decentralized *static* optimization [20], their detection and localization are much more challenging for the *dynamic* task, especially when the faulty values are time-varying.

Observe that at presence of the malfunctioning agents, at time $k$, our goal is no longer solving (1) but finding the dynamic optimal solution that minimizes the summation of the regular agents' local cost functions

$$\tilde{x}^{k*} := \arg\min_{\tilde{x}^k} \sum_{i \in \mathcal{R}} f_i^k(\tilde{x}^k). \tag{2}$$

Directly solving (2) is intractable because the identities of malfunctioning agents are not available in advance. To address this issue, we introduce a TV norm penalty on the transmitted values, which include the local iterates of the regular agents and the faulty values from the malfunctioning agents. For agent $i$, define $\mathcal{R}_i$ as the set of its regular neighbors and $\mathcal{M}_i := \mathcal{N}_i \backslash \mathcal{R}_i$ as the set of its malfunctioning neighbors. At time $k$, we expect to approximately solve

$$\begin{aligned} x^{k*} := [x_i^{k*}] = \arg \min_{x^k := [x_i^k]} & \sum_{i \in \mathcal{R}} f_i^k(x_i^k) \\ + \lambda \sum_{i \in \mathcal{R}} & \left( \frac{1}{2} \sum_{j \in \mathcal{R}_i} \|x_i^k - x_j^k\|_1 + \sum_{j \in \mathcal{M}_i} \|x_i^k - z_j^k\|_1 \right), \end{aligned} \tag{3}$$

where $x^k := [x_i^k] \in \mathbb{R}^{rp}$ is a vector that stacks all the local variables $x_i^k$ of regular agents, $x^{k*} := [x_i^{k*}] \in \mathbb{R}^{rp}$ is the optimal solution of (3), and $\lambda$ is a positive constant penalty factor. The second term in the cost function of (3) is the TV norm penalty on the transmitted values, whose minimization forces every $x_i^k$ to be close to most of the received values on agent $i$, but allows it to be different to those received outliers [21]. Therefore, when the malfunctioning agents are sparse within the network, the TV norm penalty helps mitigate their negative influence. For the applications of TV norm in identifying sparse outliers, readers are referred to [23, 24].

We propose a subgradient method to approximately solve (3) in a decentralized and dynamic manner. The subgradient of the cost function in (3) with respect to $x_i^k$ is

$$\nabla f_i^k(x_i^k) + \lambda \left( \sum_{j \in \mathcal{R}_i} sign(x_i^k - x_j^k) + \sum_{j \in \mathcal{M}_i} sign(x_i^k - z_j^k) \right),$$

where $sign(\cdot)$ is an element-wise sign function. Given $a \in \mathbb{R}$, $sign(a)$ equals to $1$ when $a > 0$, $-1$ when $a < 0$, and an arbitrary value within $[-1, 1]$ when $a = 0$. Note that this subgradient can be easily generalized to the case that $f_i^k$ is nondifferentiable, as long as we replace $\nabla f_i^k(x_i^k)$ by a subgradient of $f_i^k$ at $x_i^k$. For every regular agent $i$, its subgradient update at time $k$ is

$$\begin{aligned} x_i^{k+1} = & \, x_i^k - \alpha \nabla f_i^k(x_i^k) \\ & - \alpha\lambda \left( \sum_{j \in \mathcal{R}_i} sign(x_i^k - x_j^k) + \sum_{j \in \mathcal{M}_i} sign(x_i^k - z_j^k) \right), \end{aligned} \tag{4}$$

where $\alpha$ is a positive constant stepsize. We use a constant stepsize, other than a diminishing one, for the purpose of adapting to the dynamic cost functions [25].

The subgradient method to solve the robust decentralized dynamic optimization problem is outlined in Algorithm 1. The algorithm has two parameters, penalty factor $\lambda$ and stepsize $\alpha$. Every regular agent $i \in \mathcal{R}$ initializes its local iterate as $x_i^0$. At time $k$, it accesses the local cost function $f_i^k$, followed by receiving local iterates $x_j^k$ from regular neighbors $j \in \mathcal{R}_i$ and broadcast values $z_j^k$

from malfunctioning neighbors $j \in \mathcal{M}_i$. With all these information, it updates the local iterate $x_i^{k+1}$ according to (4). For regular agent $i$, its communication cost per iteration consists of broadcasting a $p$-dimensional vector to and receiving $|\mathcal{N}_i|$ $p$-dimensional vectors from its neighbors. The computation cost, which mainly comes from evaluating the local gradient $\nabla f_i^k(x_i^k)$, is lightweight.

**Algorithm 1** A Subgradient Method for Robust Decentralized Dynamic Optimization

**Input:** $x_i^0 \in \mathbb{R}^p$ for $i \in \mathcal{R}$, $\lambda > 0$ and $\alpha > 0$

**for** $k = 0, 1, \cdots$, every regular agent $i \in \mathcal{R}$ **do**
  Access local cost function $f_i^k$.
  Receive $x_j^k$ from regular neighbors $j \in \mathcal{R}_i$ and $z_j^k$
    from malfunctioning neighbors $j \in \mathcal{M}_i$.
  Update local iterate $x_i^{k+1}$ according to (4).
**end for**

## 3. PERFORMANCE ANALYSIS

This section analyzes the tracking performance of the proposed algorithm at presence of the malfunctioning agents. Theorem 1 investigates the TV norm regularized problem (3) at any time $k$, showing the condition under which the optimal solution of (3) is consensual and its gap from the optimal solution of (2) is bounded. Then in Theorem 2, we bound the tracking error of Algorithm 1, given that the variation of the dynamic optimal solution to (2) is bounded. All the proofs are left in a longer version [27].

We make the following assumptions on the dynamic local cost functions $f_i^k$, which are normal for convex analysis.

**Assumption 1.** ***(Lipschitz Continuous Gradients)*** *Local cost functions $f_i^k$ are differentiable and have Lipschitz continuous gradients with Lipschitz constants $M_{f_i^k} > 0$, for all regular agents $i \in \mathcal{R}$ and times $k$; namely, for any pair of points $x_i$ and $y_i$, it holds $\|\nabla f_i^k(x_i) - \nabla f_i^k(y_i)\| \leq M_{f_i^k}\|x_i - y_i\|$.*

**Assumption 2.** ***(Strong Convexity)*** *Local cost functions $f_i^k$ are strongly convex with strong convexity constants $m_{f_i^k} > 0$, for all regular agents $i \in \mathcal{R}$ and times $k$; namely, for any pair of points $x_i$ and $y_i$, it holds $[x_i - y_i]^T[\nabla f_i^k(x_i) - \nabla f_i^k(y_i)] \geq m_{f_i^k}\|x_i - y_i\|^2$.*

**Assumption 3.** ***(Bounded Gradients at Optimum)*** *Local cost functions $f_i^k$ have bounded gradients at $\tilde{x}^{k*}$, the dynamic optimal solution to (2), for all regular agents $i \in \mathcal{R}$ and times $k$; namely, $\|\nabla f_i^k(\tilde{x}^{k*})\| < \infty$.*

We also assume that the network of the regular agents is bidirectionally connected. Otherwise, consensus among regular agents is generally impossible.

**Assumption 4.** ***(Network Connectivity)*** *The network consisting of all regular agents $i \in \mathcal{R}$, denoted by $(\mathcal{R}, \mathcal{E})$, is bidirectionally connected.*

For future usage, define the node-edge incidence matrix $A = [a_{ie}] \in \mathbb{R}^{r \times |\mathcal{E}|}$ of the network with all regular agents. If $e = (i, j) \in \mathcal{E}$, then we set $a_{ie} = 1$ and $a_{je} = -1$ (the order of $i$ and $j$ is arbitrary, but by default we consider the ordered edge $(i, j)$ with $i < j$). If an agent $i$ is not attached to an edge $e$, then $a_{ie} = 0$.

The last assumption is about the variation of the dynamic optimal solution of (2) over time, which must be bounded to guarantee reasonable tracking performance.

**Assumption 5.** *For any two successive times $k-1$ and $k$, the variation of the dynamic optimal solution of (2) is bounded by a positive constant $\Theta$; namely $\|\tilde{x}^{k*} - \tilde{x}^{(k-1)*}\| \leq \Theta$.*

Before stating the main result, we introduce an auxiliary problem in the form of

$$\tilde{y}^{k*} := \arg\min_{\tilde{y}^k} \sum_{i \in \mathcal{R}} f_i^k(\tilde{y}^k) + \lambda \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{M}_i} \|\tilde{y}^k - z_j^k\|_1. \quad (5)$$

The first-order optimality condition of (5) is that, for any malfunctioning agent $j$, there exists $v_j \in \mathbb{R}^p$ whose value satisfies the definition of $sign(\tilde{y}^{k*} - z_j^k)$ such that

$$\frac{1}{\lambda} \sum_{i \in \mathcal{R}} \nabla f_i^k(\tilde{y}^{k*}) + \sum_{i \in \mathcal{R}} \sum_{j \in \mathcal{M}_i} v_j = 0. \quad (6)$$

For notational simplicity, define a vector $b^k := [b_i^k] \in \mathbb{R}^{rp}$ whose $i$th block $b_i^k := \nabla f_i^k(\tilde{y}^{k*})/\lambda + \sum_{j \in \mathcal{M}_i} v_j$. Thus, (6) is equivalent to $\sum_{i \in \mathcal{R}} b_i^k = 0$. The following theorem asserts that (3) is a good surrogate of (2), which minimizes the summation of the regular agents' local cost functions.

**Theorem 1.** *Suppose that Assumptions 1–4 hold true. Define a vector $u := [u_e] \in \mathbb{R}^{|\mathcal{E}|p}$ whose $e$th block is $u_e \in \mathbb{R}^p$. If there exists $u$ whose elements are within the range of $[-1, 1]$, such that $A \otimes I_p u + b^k = 0$ holds, then the optimal solution $x^{k*} := [x_i^{k*}]$ of (3) is consensual and the distance between every block $x_i^{k*}$ and the optimal solution $\tilde{x}^{k*}$ of (2) satisfies $\|x_i^{k*} - \tilde{x}^{k*}\| \leq \Delta^k$, where*

$$\Delta^k := \frac{\lambda\sqrt{p}}{\sum_{i \in \mathcal{R}} m_{f_i^k}} \sum_{i \in \mathcal{R}} |\mathcal{M}_i|.$$

Note that [21] analyzes conditions under which (3) achieves consensus, as well as (2) and (3) are equivalent, given that all the agents are regular. Here we extend the result to the case that the malfunctioning agents exist and play negative roles. In addition, [21] considers the *static* TV norm regularized problem, while we further investigate the *dynamic* tracking performance, as shown below.

We further show that Algorithm 1, which approximately solves (3) using one subgradient evaluation per time, other than running an inner loop of multiple subgradient evaluations at each time index, tracks the optimal solution of (3) with bounded error.

**Theorem 2.** *Given that the conditions of Theorem 1 as well as Assumption 5 hold true, the tracking error of Algorithm 1 is upper bounded by*

$$\begin{aligned} \|x^k - x^{k*}\| \leq & c^k \|x^0 - x^{0*}\| \\ & + \frac{1}{1-c}\left(2c\sqrt{r}\max_k \Delta^k + c\sqrt{r}\Theta + d\right), \end{aligned} \quad (7)$$

*if the stepsize $\alpha < 1/(\min_{i \in \mathcal{R}} m_{f_i^k} + \max_{i \in \mathcal{R}} M_{f_i^k})$. Here $c := (1 - 2\alpha m_{f^k} M_{f^k}/(m_{f^k} + M_{f^k}))^{1/2}$, $d := (8\alpha^2\lambda^2 p \sum_{i \in \mathcal{R}} |\mathcal{N}_i|^2)^{1/2}$ are two constants.*

Since $c$ is a constant within the range of $(0, 1)$, Theorem 2 implies that the influence of the initial tracking error $\|x^0 - x^{0*}\|$ vanishes at an exponential rate. The steady-state tracking error, as $k \to \infty$, is proportional to $\max_k \Delta^k$ (the gap between the optimal solutions of (2) and (5)), $\Theta$ (the variation of the dynamic optimal solution of (2)), as well as $(\sum_{i \in \mathcal{R}} |\mathcal{N}_i|^2)^{1/2}$ (a constant determined by the topology of regular and malfunctioning agents).

Theorem 2 shows how Algorithm 1 tracks the optimal solution of (3). Combining Theorems 1 and 2, it is straightforward assert that

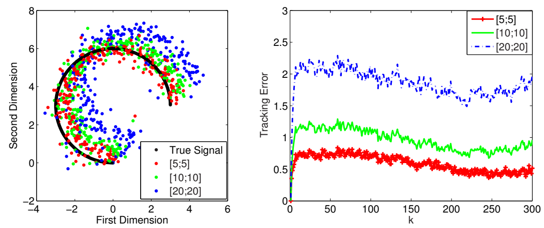


**Fig. 1**. DGD with three malfunctioning agents broadcasting the same faulty vectors, [5; 5], [10; 10] or [20; 20].

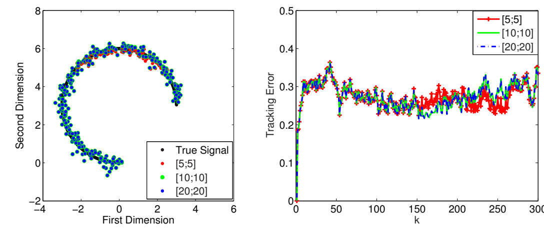


**Fig. 2**. Algorithm 1 with three malfunctioning agents broadcasting the same faulty vectors, [5; 5], [10; 10] or [20; 20].

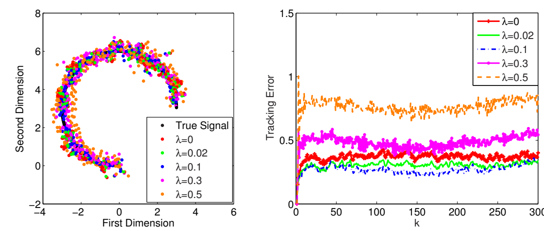


**Fig. 3**. Algorithm 1 with three malfunctioning agents broadcasting the same faulty vectors [10; 10].

Algorithm 1 is also able to track the dynamic optimal solution of (2) with bounded error

$$\|x^k - [\tilde{x}^{k*}]\| \le c^k \|x^0 - x^{0*}\| + \frac{1}{1-c}\left((1+c)\sqrt{r}\max_k \Delta^k + c\sqrt{r}\Theta + d\right). \tag{8}$$

Here $[\tilde{x}^{k*}] \in \mathbb{R}^{rp}$ stacks $r$ optimal solution $\tilde{x}^{k*}$ of (2).

## 4. NUMERICAL EXPERIMENTS

We compare the proposed algorithm with the dynamic version of the celebrated decentralized gradient descent (DGD) method [26], which does not consider mitigating the influence of the malfunctioning agents. At time $k$, DGD updates the local variables as

$$x_i^{k+1} = \sum_{j \in \mathcal{R}_i} w_{ij} x_j^k + \sum_{j \in \mathcal{M}_i} w_{ij} z_j^k - \beta \nabla f_i(x_i^k), \tag{9}$$

for all $i \in \mathcal{R}$. Here $\beta$ is a positive constant stepsize, and $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ is the mixing matrix of the entire network including both regular and malfunctioning agents. In the numerical experiments, we choose $W$ according to the maximum-degree rule [28].

We consider a random geometric graph, which uniformly randomly places 100 agents in a two-dimensional area $[0, 3] \times [0, 3]$ and treats two agents as neighbors if and only if their distance is less than 1. The agents track a moving target whose true position $\check{x}^k \in \mathbb{R}^2$ evolves along a $3/4$ circle, starting from $(0, 0)$, heading to $(-3, 3)$ and then $(6, 0)$, and ending at $(3, 3)$. The velocity of the target is constant and each $1/4$ circle takes 100 time slots. At time $k$, regular agent $i$ measures a true position $\check{x}^k$ through a linear observation function $y_i^k = H_i^k \check{x}^k + e_i^k$, where elements of the measurement matrix $H_i^k \in \mathbb{R}^{2\times 2}$ follow normal distribution $\mathcal{N}(0, 1)$ and elements of the measurement noise $e_i^k \in \mathbb{R}^2$ follow normal distribution $\mathcal{N}(0, 1)$. Thus, the regular agents aim at finding $\tilde{x}^{k*} := \arg\min \sum_{i \in \mathcal{R}} f_i^k(\tilde{x}^k)$, where $f_i^k(\tilde{x}^k) = \|H_i^k \tilde{x}^k - y_i^k\|^2/2$. The performance metric is tracking error defined by $\sum_{i \in \mathcal{R}} \|x_i^k - \tilde{x}^{k*}\|/r$.

**Comparison with DGD**. Randomly choose $m = 3$ malfunctioning agents among $n = 100$ agents, but guarantee that the network of regular agents is connected. Suppose that the malfunctioning agents broadcast the same faulty vectors. We consider three different settings for faulty vectors: for all $k$ and for all $i \in \mathcal{M}$, $z_i^k = [5; 5]$, $z_i^k = [10; 10]$, or $z_i^k = [20; 20]$. Performance of DGD with stepsize $\beta = 0.2$, which is hand-tuned to yield balanced tracking performance, is illustrated in Fig. 1. The left plot compares the true signal $\check{x}^k$ and the decentralized estimates of a randomly chosen regular agent for different levels of faulty values. When the magnitude of the faulty vectors becomes larger, the bias between the decentralized estimate and the true signal also becomes more significant. The impact of the faulty vectors can be further observed in the right plot, which shows the overall tracking error of the network. As the faulty vectors vary from $[5, 5]$ to $[20, 20]$, the steady-state tracking error increases from around $0.5$ to around $2$. Performance of Algorithm 1 with stepsize $\alpha = 0.1$ and regularization parameter $\lambda = 0.1$ is illustrated in Fig. 2. Thanks to the TV norm regularization term, the network is not sensitive to the faulty vectors broadcast by the malfunctioning agents. The left plot shows that, no matter how the faulty vectors vary, the decentralized estimate of a randomly chosen regular agent is always close to the true signal $\check{x}^k$. The right plot depicts that the steady-state tracking errors are always around $0.3$, which are much smaller than those of DGD, for all the three cases.

**Impact of Regularization Parameter $\lambda$.** Randomly choose $m = 3$ malfunctioning agents among $n = 100$ agents and suppose that the malfunctioning agents broadcast the same faulty vectors $[10; 10]$. The stepsize remains to be $\alpha = 0.1$ but the regularization parameter $\lambda$ varies from 0, 0.02, 0.1, 0.3 to 0.5. Note that $\lambda = 0$ corresponds to that the malfunctioning agents do not collaborate with any others, no matter regular or malfunctioning, and independently optimize their own local cost functions. The left plot of Fig. 3 shows the decentralized estimates of a random regular agent, which are not far away from the true signal and are robust to the setting of $\lambda$. Observing the right plot, we can see that too large or too small $\lambda$ both yield large steady-state tracking error. Note that a large $\lambda$ helps consensus of the regular agents, but the reached consensus is not necessarily close to the dynamic optimal solution. On the other hand, a small $\lambda$ allows the regular agents to be "selfish" such that network-wide consensus is often violated. The numerical experiments demonstrate that setting a proper $\lambda$ helps achieve the tradeoff between consensus and approximation accuracy.

**Acknowledgement**. This work is supported in part by the China National Science Foundation under Grant 61573331 and the Anhui Provincial Natural Science Foundation under Grant 1608085QF130.

## 5. REFERENCES


[1] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," IEEE Transactions on Signal Processing, vol. 62, no. 5, pp. 1185–1197, 2014.

[2] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "Decentralized prediction-correction methods for networked time-varying convex optimization," IEEE Transactions on Signal Processing, vol. 64, no. 17, pp. 4576–4591, 2016.

[3] M. Fazlyab, S. Paternain, V. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," arXiv: 1608.07544, 2016.

[4] S. Tu and A. Sayed, "Mobile adaptive networks," IEEE Journal of Selected Topics in Signal Processing, vol. 5, no. 4, pp. 649–664, 2011.

[5] F. Jakubiec and A. Ribeiro, "D-MAP: Distributed maximum a posteriori probability estimation of dynamic systems," IEEE Transactions on Signal Processing, vol. 61, no. 2, pp. 450–466, 2013.

[6] R. Cavalcante and S. Stanczak, "A distributed subgradient method for dynamic convex optimization problems under noisy information exchange," IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 2, pp. 243–256, 2013.

[7] K. Zhou and S. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," IEEE Transactions on Robotics, vol. 27, no. 4, pp. 678–695, 2010.

[8] M. Ye and G. Hu, "Distributed optimization for systems with time-varying quadratic objective functions," In: Proceedings of CDC, 2015.

[9] S. Rahili and W. Ren, "Distributed continuous-time convex optimization with time-varying cost functions," IEEE Transactions on Automatic Control, vol. 62, no. 4, pp. 1590–1605, 2017.

[10] C. Enyioha, A. Jadbabaie, V. Preciado, and G. Pappas, "Distributed resource allocation for epidemic control," arXiv: 1501.01701, 2015.

[11] T. Chen, Q. Ling, and G. Giannakis, "An online convex optimization approach to dynamic network resource allocation," arXiv: 1701.03974, 2017.

[12] M. Maros and J. Jalden, "ADMM for distributed dynamic beam-forming," IEEE Transactions on Signal and Information Processing over Networks, 2017.

[13] H. Liu, W. Shi, and H. Zhu, "Decentralized dynamic optimization for power network voltage control," IEEE Transactions on Signal and Information Processing over Networks, 2017.

[14] C. Enyioha, S. Magnusson, K. Heal, N. Li, C. Fischione, and V. Tarokh, "On variability of renewable energy and online power allocation," IEEE Transactions on Power Systems, 2017.

[15] B. Kailkhura, S. Brahma, and P. Varshney, "Data falsification attacks on consensus-based detection systems," IEEE Transactions on Signal and Information Processing over Networks, vol. 3, no. 1, pp. 145–158, 2017.

[16] E. Nurellari, D. McLernon, and M. Ghogho, "A secure optimum distributed detection scheme in under-attack wireless sensor networks," IEEE Transactions on Signal and Information Processing over Networks, 2017.

[17] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," arXiv: 1606.08939, 2016.

[18] D. Acemoglu, G. Como, F. Fagnani, and A. Ozdaglar, "Opinion fluctuations and disagreement in social networks," Mathematics of Operations Research, vol. 38, no. 1, pp. 1–27, 2013.

[19] E. Yildiz, A. Ozdaglar, D. Acemoglu, and A. Scaglione, "Binary opinion dynamics with stubborn agents," ACM Transactions on Economics and Computation, vol. 1, no. 4, article 19, 2013.

[20] R. Gentz, S. Wu, H. Wai, A. Scaglione, and A. Leshem, "Data injection attacks in randomized gossiping," IEEE Transactions on Signal and Information Processing over Networks, vol. 2, no. 4, pp. 523–538, 2016.

[21] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, "Robust distributed consensus using total variation," IEEE Transactions on Automatic Control, vol. 61, no. 6, pp. 1550–1564, 2016.

[22] A. Koppel, B. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," IEEE Transactions on Signal Processing, vol. 65, no. 12, pp. 3062–3077, 2017.

[23] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D: Nonlinear Phenomena, vol. 60, no. 1–4, pp. 259–268, 1992.

[24] A. Chambolle, "An algorithm for total variation minimization and applications," Journal of Mathematical Imaging and Vision, vol. 20, no. 1–2, pp. 89–97, 2004.

[25] A. Sayed, "Adaptive networks," Proceedings of the IEEE, vol. 102, no. 4, pp. 460–497, 2014.

[26] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," IEEE Transactions on Automatic Control, vol. 54, no. 1, pp. 48–61, 2009.

[27] W. Xu, Z. Li, and Q. Ling, "Robust decentralized dynamic optimization at presence of malfunctioning agents," Available Online: home.ustc.edu.cn/~qingling/pdf/rddo-submitted.pdf

[28] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," SIAM Review, vol. 46, no. 4, pp. 667–689, 2004.