

A CONTINUOUS-TIME DISTRIBUTED ALGORITHM FOR SOLVING A CLASS OF DECOMPOSABLE NONCONVEX QUADRATIC PROGRAMMING

Yan Zhao¹, Qingshan Liu^{2*}

¹*School of Common Courses, Wannan Medical College
Wuhu 241000, China*

²*School of Mathematics, Southeast University
Nanjing 210096, China*

Submitted: 16th January 2018; Accepted: 26th March 2018

Abstract

In this paper, a continuous-time distributed algorithm is presented to solve a class of decomposable quadratic programming problems. In the quadratic programming, even if the objective function is nonconvex, the algorithm can still perform well under an extra condition combining with the objective, constraint and coupling matrices. Inspired by recent advances in distributed optimization, the proposed continuous-time algorithm described by multi-agent network with consensus is designed and analyzed. In the network, each agent only accesses the local information of its own and from its neighbors, then all the agents in a connected network cooperatively find the optimal solution with consensus.

Keywords: Decomposable nonconvex quadratic programming, multi-agent network, consensus, Lyapunov method

1 Introduction

In the past years, the study on optimization and applications has brought an increasing interest in science and engineering and many optimization methods have been investigated, such as interior point algorithm [1], gradient projection method [2], neurodynamic method [3, 4, 5, 6], swarm intelligence method [7, 8, 9], and distributed algorithms [10, 11, 12]. In contrast to centralized algorithms, distributed algorithms give an efficient computing method for large-scale optimization problems. Especially, the distributed algorithms based on multi-agent networks can be analyzed by the theories of dynamic systems and graph analysis method. Multi-agent network is often described as

connected directed or undirected graphs [13, 14], which dynamic behaviors are described as ordinary differential equations or difference equations. The connection properties can be analyzed using the algebraic graph theory and the dynamic properties can be analyzed using the Lyapunov method [15, 16, 17]. The goal of multi-agent networks for distributed optimization is to minimize or maximize a separable objective function with coupling or decoupling constraints. In the network, each agent gets the information from its neighbors combining with its own objective and constraints, then all the agents synchronously update their states by these information. At the same time, all the agents cooperatively get optimal solutions with consensus or

*Corresponding author (email: qslu@seu.edu.cn). This work was supported in part by the National Natural Science Foundation of China under Grant 61473333.

agreement. In general, the consensus is required for distributed algorithms with primal variables, dual variables or both of them. The applications of distributed optimization can be found in many real problems, including sensor fusion [18], distributed model predictive control and network flows [19], distributed coordination in multi-vehicle cooperative control [20], etc.

Many distributed algorithms have been designed for different optimization problems. In [13, 14, 21], based on the primal-dual method, the distributed subgradient algorithms for constrained convex optimization are presented and the average consensus is analyzed. Based on stochastic transition matrix, the distributed multi-agent optimization with nonidentical constraints investigated in [22, 23]. The robust distributed optimization is presented in [24] based on the cutting-plane consensus algorithm. The distributed random convex programming with a large number of randomly extracted constraints is presented in [25]. Based on directed graph method, the unconstrained convex optimization based on multi-agent network with weight-balanced digraphs is investigated in [16]. With the presence of external disturbances, the authors in [26] investigate the multi-agent networks for distributed optimization and the exact optimal solutions can be guaranteed with rejecting disturbances. In [27] and [28], the multi-agent networks with delays are investigated for continuous-time and discrete-time systems respectively. In [17], a second-order multi-agent network is proposed for solving distributed optimization problems with bound constraints, which optimal solutions can be exactly achieved for connected and undirected graphs. In [29], a collective neurodynamic approach, which is also described as distributed neurodynamic system, is presented for distributed optimization with decoupling constraints.

This paper is concerned with continuous-time distributed algorithm based on multi-agent network for solving decomposable quadratic programming problems. First, the distributed algorithm is described as a continuous-time multi-agent system based on the optimality conditions. Then, the consensus on primal variables is investigated for the multi-agent network to guarantee the optimal solutions. It is noted that the distributed algorithm is capable of solving a class of nonconvex quadratic

programming problems. Compared with other distributed algorithms, the coupling matrix plays an important role in the nonconvex optimization.

The remainder of the paper are listed as follows. Section 2 introduces the problem formulation and optimality conditions. In Section 3, the proposed distributed algorithm is given and the consensus is analyzed. Next, Section 4 provides an illustrative example to show the performance of the distributed algorithm. Finally, Section 5 is the conclusion of the paper.

2 Distributed Optimization and Optimality Condition

2.1 Problem Formulation

Consider the constrained decomposable quadratic programming problem as

$$\begin{aligned} & \text{minimize } \sum_{i=1}^m f_i(y), \\ & \text{subject to } C_i y = d_i, \quad i = 1, 2, \dots, m, \\ & \quad y \in \bigcap_{i=1}^m \Omega_i, \end{aligned} \quad (1)$$

where $y \in \mathbb{R}^n$, $f_i : y^T Q_i y / 2 + q_i^T y$ with $Q_i \in \mathbb{R}^{n \times n}$ to be symmetric but not necessarily to be positive semi-definite, $q_i \in \mathbb{R}^n$, $C_i \in \mathbb{R}^{s_i \times n}$, $d_i \in \mathbb{R}^{s_i}$ ($0 \leq s_i < n$), and $\Omega_i \subset \mathbb{R}^n$ is nonempty and closed convex set. Throughout the paper, we always assume that there exists at least one finite solution to problem (1).

2.2 Equivalent Optimization Problem

Since the problem in (1) is divided into m sub-problems with decomposed objective functions and constraints, the optimization problem is required to be solved using m agents under a connected and undirected graph, which is denoted by \mathcal{G} . In the network, each agent has its own local objective function $f_i(y)$ and constraint $C_i y = d_i$ with $y \in \Omega_i$.

Assume $y^i \in \mathbb{R}^n$ to be the estimate of agent i on the solution of problem (1), matrix $Y = (y^1, y^2, \dots, y^m) \in \mathbb{R}^{n \times m}$ to be with column vector y^i , and column vector $\tilde{y} = \text{vec}(Y) \in \mathbb{R}^{mn}$ to be the vectorization of Y .

Lemma 1 [16] Assume $L_m \in \mathbb{R}^{m \times m}$ to be the Laplacian matrix of the graph and $L = L_m \otimes I_n \in \mathbb{R}^{mn \times mn}$, in which I_n is n -dimensional identity ma-

trix and \otimes is the Kronecker product. Then (1) is equivalent to the following programming problem if the graph is connected

$$\begin{aligned} & \text{minimize } f(\tilde{y}), \\ & \text{subject to } C\tilde{y} = d, \\ & \quad L\tilde{y} = 0, \\ & \quad \tilde{y} \in \Omega, \end{aligned} \quad (2)$$

where $f(\tilde{y}) = \sum_{i=1}^m f_i(y^i)$, C is the block diagonal matrix of C_1, C_2 to C_m (i.e., $C = \text{blkdiag}\{C_1, C_2, \dots, C_m\} \in \mathbb{R}^{s \times mn}$), $d = \text{vec}(d_1, d_2, \dots, d_m) \in \mathbb{R}^s$, and $\Omega = \prod_{i=1}^m \Omega_i$, in which $s = s_1 + s_2 + \dots + s_m$.

2.3 Optimality Condition

To get the optimal solutions of (2), a necessary condition is given in the following theorem.

Theorem 2 *If $\tilde{y}^* \in \mathbb{R}^{mn}$ is an optimal solution to (2), there exist $\tilde{v}^* \in \mathbb{R}^s$ and $\tilde{w}^* \in \mathbb{R}^{mn}$ such that $(\tilde{y}^*, \tilde{v}^*, \tilde{w}^*)$ satisfies*

$$\begin{cases} \tilde{y}^* = \phi(\tilde{y}^* - \nabla f(\tilde{y}^*) - C^T \tilde{v}^* - L\tilde{w}^*), \\ C\tilde{y}^* = d, \\ L\tilde{y}^* = 0, \end{cases} \quad (3)$$

where ϕ is a projection operator from \mathbb{R}^{mn} to Ω defined in Appendix.

Proof The Lagrange function of (2) is

$$\Phi(\tilde{y}, \tilde{v}, \tilde{w}) = f(\tilde{y}) + \tilde{v}^T (C\tilde{y} - d) + \tilde{w}^T L\tilde{y},$$

where $\tilde{v}^* \in \mathbb{R}^s$ and $\tilde{w}^* \in \mathbb{R}^{mn}$ are the Lagrange multipliers. From the Karush-Kuhn-Tucker (KKT) conditions [30], if \tilde{y}^* is an optimal solution to (2), then $(\tilde{y}^*, \tilde{v}^*, \tilde{w}^*)$ satisfies

$$(\tilde{y} - \tilde{y}^*)^T (\nabla f(\tilde{y}^*) + C^T \tilde{v}^* + L\tilde{w}^*) \geq 0, \quad \forall \tilde{y} \in \Omega, \quad (4)$$

$$C\tilde{y}^* = d, \quad (5)$$

$$L\tilde{y}^* = 0, \quad (6)$$

It is well known that \tilde{y}^* is a solution to the variational inequality (4) if and only if \tilde{y}^* satisfies [31]

$$\tilde{y}^* = \phi(\tilde{y}^* - \nabla f(\tilde{y}^*) - C^T \tilde{v}^* - L\tilde{w}^*).$$

Thus the equations in (3) are satisfied.

Remark 2.1 *In (2), if $f(\tilde{y})$ is convex, which is equivalent to the matrix $Q = \text{blkdiag}\{Q_1, Q_2, \dots, Q_m\}$ being positive semi-definite, the condition in (3) is also sufficient.*

3 Continuous-time Distributed Algorithm and Consensus Analysis

Next, based on the equations in (3), a continuous-time multi-agent system is proposed to solve (1) via distributed manner.

3.1 Model Description

From (3), the dynamic system of the continuous-time multi-agent network is described as

$$\begin{cases} \frac{d\tilde{y}}{dt} = 2[-\tilde{y} + \phi(\tilde{y} - \nabla f(\tilde{y}) - C^T(\tilde{v} + C\tilde{y} - d) - \tilde{z} - L\tilde{y})], \\ \frac{d\tilde{v}}{dt} = C\tilde{y} - d, \\ \frac{d\tilde{z}}{dt} = L\tilde{y}. \end{cases} \quad (7)$$

It is obvious that if $(\tilde{y}^*, \tilde{v}^*, \tilde{w}^*)$ satisfies the equations in (3), it is also an equilibrium point of system (7) with $\tilde{z}^* = L\tilde{w}^*$ and vice versa.

For $i = 1, 2, \dots, m$, the component form of system (7) can be written as

$$\begin{cases} \frac{dy^i}{dt} = 2[-y^i + \phi_i(y^i - \nabla f_i(y^i) - C_i^T(v^i + C_i y^i - d_i) - z^i - \sum_{j=1, j \neq i}^m a_{ij}(y^i - y^j))], \\ \frac{dv^i}{dt} = C_i y^i - d_i, \\ \frac{dz^i}{dt} = \sum_{j=1, j \neq i}^m a_{ij}(y^i - y^j), \end{cases} \quad (8)$$

where a_{ij} is the connection weight between agents i and j in the network.

From system (8), we can see that each agent is designed to solve a local problem with objective and constraints. Meanwhile, each agent exchanges its information on y^i with its neighbors. Then all the agents synchronously update their information and cooperatively solve the problem with consensus.

3.2 Consensus Analysis for Indefinite Q

To prove the convergence of the system in (7), another system is introduced as follows

$$\begin{cases} \frac{d\tilde{y}}{dt} = 2[-\tilde{y} + \phi(\tilde{y} - \nabla f(\tilde{y}) \\ \quad - C^T(\tilde{v} + C\tilde{y} - d) - L(\tilde{y} + \tilde{w}))], \\ \frac{d\tilde{v}}{dt} = C\tilde{y} - d, \\ \frac{d\tilde{w}}{dt} = \tilde{y}. \end{cases} \quad (9)$$

We say that the systems in (7) and (9) are equivalent according to the following lemma.

Lemma 3 [32] *The systems in (7) and (9) are equivalent if the initial conditions satisfy $\tilde{z}(0) = L\tilde{w}(0)$ for any $\tilde{w}(0) \in \mathbb{R}^{mn}$.*

In the following, the convergence of system (9) will be proved. Thus the results are also true for the system in (7).

Theorem 4 *The state vectors $y^i(t)$ ($i = 1, 2, \dots, m$) of system (9) reach consensus at the unique optimal solution of (1) if $2Q + C^T C + L$ is positive definite and the graph of the multi-agent network is connected and undirected.*

Proof Assume $\tilde{y}^* \in \mathbb{R}^{mn}$ to be an optimal solution of (2). From Theorem 2, there exist $\tilde{v}^* \in \mathbb{R}^s$ and $\tilde{w}^* \in \mathbb{R}^{mn}$ such that the formulas in (3) hold.

We construct the candidate Lyapunov function as

$$\begin{aligned} V(\tilde{y}, \tilde{v}, \tilde{w}) &= \varphi(\tilde{y}, \tilde{v}, \tilde{w}) - \varphi(\tilde{y}^*, \tilde{v}^*, \tilde{w}^*) - (\tilde{y} - \tilde{y}^*)^T (\nabla f(\tilde{y}^*) \\ &\quad + C^T \tilde{v}^* + L\tilde{w}^*) - (\tilde{v} - \tilde{v}^*)^T \tilde{v}^* \\ &\quad - (\tilde{w} - \tilde{w}^*)^T L\tilde{w}^* + \frac{1}{2} (\|\tilde{y} - \tilde{y}^*\|^2 \\ &\quad + \|\tilde{v} - \tilde{v}^*\|^2 + (\tilde{w} - \tilde{w}^*)^T L(\tilde{w} - \tilde{w}^*)), \end{aligned} \quad (10)$$

where

$$\begin{aligned} \varphi(\tilde{y}, \tilde{v}, \tilde{w}) &= f(\tilde{y}) + \frac{1}{2} (\|\tilde{v} + C\tilde{y} - d\|^2 \\ &\quad + (\tilde{y} + \tilde{w})^T L(\tilde{y} + \tilde{w})). \end{aligned}$$

We have

$$\begin{aligned} \nabla \varphi(\tilde{y}, \tilde{v}, \tilde{w}) &= \begin{pmatrix} \nabla f(\tilde{y}) + C^T(\tilde{v} + C\tilde{y} - d) + L(\tilde{y} + \tilde{w}) \\ \tilde{v} + C\tilde{y} - d \\ L(\tilde{y} + \tilde{w}) \end{pmatrix}, \end{aligned}$$

The gradient of $V(\tilde{y}, \tilde{v}, \tilde{w})$ is shown as

$$\begin{aligned} \nabla V(\tilde{y}, \tilde{v}, \tilde{w}) &= \begin{pmatrix} \nabla f(\tilde{y}) + C^T(\tilde{v} + C\tilde{y} - d) \\ + L(\tilde{y} + \tilde{w}) - \nabla f(\tilde{y}^*) \\ - C^T \tilde{v}^* - L\tilde{w}^* + \tilde{y} - \tilde{y}^* \\ 2(\tilde{v} - \tilde{v}^*) + C\tilde{y} - d \\ 2L(\tilde{w} - \tilde{w}^*) + L\tilde{y} \end{pmatrix}. \end{aligned} \quad (11)$$

Then the derivative of $V(\tilde{y}, \tilde{v}, \tilde{w})$ along the solution of (9) is

$$\begin{aligned} \dot{V}(\tilde{y}(t), \tilde{v}(t), \tilde{w}(t)) &= \nabla_{\tilde{y}} V(\tilde{y}, \tilde{v}, \tilde{w})^T \left(\frac{d\tilde{y}}{dt} \right) + \nabla_{\tilde{v}} V(\tilde{y}, \tilde{v}, \tilde{w})^T \left(\frac{d\tilde{v}}{dt} \right) \\ &\quad + \nabla_{\tilde{w}} V(\tilde{y}, \tilde{v}, \tilde{w})^T \left(\frac{d\tilde{w}}{dt} \right). \end{aligned}$$

Then

$$\begin{aligned} \dot{V}(\tilde{y}(t), \tilde{v}(t), \tilde{w}(t)) &= 2(\nabla f(\tilde{y}) + C^T(\tilde{v} + C\tilde{y} - d) + L(\tilde{y} + \tilde{w}) \\ &\quad - \nabla f(\tilde{y}^*) - C^T \tilde{v}^* - L\tilde{w}^* + \tilde{y} - \tilde{y}^*)^T (-\tilde{y} + \tilde{\phi}) \\ &\quad + (2(\tilde{v} - \tilde{v}^*) + C\tilde{y} - d)^T (C\tilde{y} - d) \\ &\quad + (2L(\tilde{w} - \tilde{w}^*) + L\tilde{y})^T \tilde{y}, \end{aligned} \quad (12)$$

where $\tilde{\phi} = \phi(\tilde{y} - \nabla f(\tilde{y}) - C^T(\tilde{v} + C\tilde{y} - d) - L(\tilde{y} + \tilde{w}))$.

For the right hand side of (12), we have

$$\begin{aligned} &2(\nabla f(\tilde{y}) + C^T(\tilde{v} + C\tilde{y} - d) + L(\tilde{y} + \tilde{w}) \\ &\quad - \nabla f(\tilde{y}^*) - C^T \tilde{v}^* - L\tilde{w}^* + \tilde{y} - \tilde{y}^*)^T (-\tilde{y} + \tilde{\phi}) \\ &\quad + (2(\tilde{v} - \tilde{v}^*) + C\tilde{y} - d)^T (C\tilde{y} - d) \\ &\quad + (2L(\tilde{w} - \tilde{w}^*) + L\tilde{y})^T \tilde{y} \\ &= 2(\nabla f(\tilde{y}) + C^T(\tilde{v} + C\tilde{y} - d) + L(\tilde{y} + \tilde{w}) \\ &\quad - \nabla f(\tilde{y}^*) - C^T \tilde{v}^* - L\tilde{w}^* + \tilde{y} - \tilde{y}^*)^T (\tilde{\phi} - \tilde{y}^*) \\ &\quad + 2(\nabla f(\tilde{y}) + C^T(\tilde{v} + C\tilde{y} - d) + L(\tilde{y} + \tilde{w}) \\ &\quad - \nabla f(\tilde{y}^*) - C^T \tilde{v}^* - L\tilde{w}^* + \tilde{y} - \tilde{y}^*)^T (\tilde{y}^* - \tilde{y}) \\ &\quad + (2(\tilde{v} - \tilde{v}^*) + C\tilde{y} - d)^T (C\tilde{y} - d) \\ &\quad + 2(L(\tilde{w} - \tilde{w}^*))^T \tilde{y} + \tilde{y}^T L\tilde{y}. \end{aligned}$$

Let

$$\begin{aligned} J_1 &= 2(\nabla f(\tilde{y}) + C^T(\tilde{v} + C\tilde{y} - d) \\ &\quad + L(\tilde{y} + \tilde{w}) - \nabla f(\tilde{y}^*) - C^T \tilde{v}^* \\ &\quad - L\tilde{w}^* + \tilde{y} - \tilde{y}^*)^T (\tilde{\phi} - \tilde{y}^*), \end{aligned}$$

$$J_2 = 2(\nabla f(\tilde{y}) + L(\tilde{y} + \tilde{w}) - \nabla f(\tilde{y}^*) - L\tilde{w}^* + \tilde{y} - \tilde{y}^*)^T (\tilde{y}^* - \tilde{y}) + 2(L(\tilde{w} - \tilde{w}^*))^T \tilde{y} + \tilde{y}^T L\tilde{y},$$

and

$$J_3 = 2(C^T(\tilde{v} + C\tilde{y} - d) - C^T\tilde{v}^*)^T (\tilde{y}^* - \tilde{y}) + (2(\tilde{v} - \tilde{v}^*) + C\tilde{y} - d)^T (C\tilde{y} - d).$$

Then

$$\dot{V}(\tilde{y}(t), \tilde{v}(t), \tilde{w}(t)) = J_1 + J_2 + J_3. \quad (13)$$

For J_1 , we have

$$\begin{aligned} J_1 &= -2(\tilde{y} - \nabla f(\tilde{y}) - C^T(\tilde{v} + C\tilde{y} - d) - L(\tilde{y} + \tilde{w}) - \tilde{\phi})^T (\tilde{\phi} - \tilde{y}^*) \\ &\quad - 2(\nabla f(\tilde{y}^*) + C^T\tilde{v}^* + L\tilde{w}^*)^T (\tilde{\phi} - \tilde{y}^*) \\ &\quad + 2(\tilde{y} - \tilde{\phi} + \tilde{y} - \tilde{y}^*)^T (\tilde{\phi} - \tilde{y} + \tilde{y} - \tilde{y}^*) \\ &= -2(\tilde{y} - \nabla f(\tilde{y}) - C^T(\tilde{v} + C\tilde{y} - d) - L(\tilde{y} + \tilde{w}) - \tilde{\phi})^T (\tilde{\phi} - \tilde{y}^*) \\ &\quad - 2(\nabla f(\tilde{y}^*) + C^T\tilde{v}^* + L\tilde{w}^*)^T (\tilde{\phi} - \tilde{y}^*) \\ &\quad - 2\|\tilde{y} - \tilde{\phi}\|^2 + 2\|\tilde{y} - \tilde{y}^*\|^2. \end{aligned}$$

From Lemma 6 in Appendix, let $\omega = \tilde{y} - \nabla f(\tilde{y}) - C^T(\tilde{v} + C\tilde{y} - d) - L(\tilde{y} + \tilde{w})$ and $\mathbf{v} = \tilde{y}^*$, then

$$\begin{aligned} &(\tilde{y} - \nabla f(\tilde{y}) - C^T(\tilde{v} + C\tilde{y} - d) - L(\tilde{y} + \tilde{w}) - \tilde{\phi})^T (\tilde{\phi} - \tilde{y}^*) \\ &\geq 0. \end{aligned}$$

Since \tilde{y}^* is a solution of (2), the inequality in (4) holds. It follows that $(\nabla f(\tilde{y}^*) + C^T\tilde{v}^* + L\tilde{w}^*)^T (\tilde{\phi} - \tilde{y}^*) \geq 0$ due to $\tilde{\phi} \in \Omega$.

Then

$$J_1 \leq -2\|\tilde{y} - \tilde{\phi}\|^2 + 2\|\tilde{y} - \tilde{y}^*\|^2.$$

For J_2 , we have

$$\begin{aligned} J_2 &= 2(\nabla f(\tilde{y}) + L(\tilde{y} + \tilde{w}) - \nabla f(\tilde{y}^*) - L\tilde{w}^* + \tilde{y} - \tilde{y}^*)^T (\tilde{y}^* - \tilde{y}) \\ &\quad + 2(L(\tilde{w} - \tilde{w}^*))^T \tilde{y} + \tilde{y}^T L\tilde{y} \\ &= 2(\nabla f(\tilde{y}) - \nabla f(\tilde{y}^*))^T (\tilde{y}^* - \tilde{y}) \\ &\quad + 2(L(\tilde{y} + \tilde{w}) - L\tilde{w}^*)^T (\tilde{y}^* - \tilde{y}) \\ &\quad + 2(\tilde{y} - \tilde{y}^*)^T (\tilde{y}^* - \tilde{y}) \\ &\quad + 2(L(\tilde{w} - \tilde{w}^*))^T \tilde{y} + \tilde{y}^T L\tilde{y} \\ &= 2(\nabla f(\tilde{y}) - \nabla f(\tilde{y}^*))^T (\tilde{y}^* - \tilde{y}) \\ &\quad + 2(\tilde{y} - \tilde{y}^*)^T (\tilde{y}^* - \tilde{y}) - \tilde{y}^T L\tilde{y}, \end{aligned}$$

in which the last equality holds due to $L\tilde{y}^* = 0$.

Since

$$\begin{aligned} f(\tilde{y}) &= \sum_{i=1}^m f_i(y^i) \\ &= \sum_{i=1}^m \left(\frac{1}{2} (y^i)^T Q_i y^i + q_i^T y^i \right), \end{aligned}$$

we have

$$\begin{aligned} &(\tilde{y} - \tilde{y}^*)^T (\nabla f(\tilde{y}) - \nabla f(\tilde{y}^*)) \\ &= \sum_{i=1}^m (y^i - (y^i)^*)^T Q_i (y^i - (y^i)^*) \\ &= (\tilde{y} - \tilde{y}^*)^T Q (\tilde{y} - \tilde{y}^*). \end{aligned}$$

Then

$$\begin{aligned} J_2 &= -2(\tilde{y} - \tilde{y}^*)^T Q (\tilde{y} - \tilde{y}^*) \\ &\quad - 2\|\tilde{y} - \tilde{y}^*\|^2 - \tilde{y}^T L\tilde{y}. \end{aligned}$$

For J_3 , since $C\tilde{y}^* = d$, we have

$$\begin{aligned} J_3 &= 2(\tilde{v} - \tilde{v}^* + C\tilde{y} - d)^T (d - C\tilde{y}) \\ &\quad + (2(\tilde{v} - \tilde{v}^*) + C\tilde{y} - d)^T (C\tilde{y} - d) \\ &= -\|C\tilde{y} - d\|^2. \end{aligned}$$

Consequently, combining with (13) follows

$$\begin{aligned} &\dot{V}(\tilde{y}(t), \tilde{v}(t), \tilde{w}(t)) \\ &= J_1 + J_2 + J_3 \\ &\leq -2\|\tilde{y} - \tilde{\phi}\|^2 - 2(\tilde{y} - \tilde{y}^*)^T Q (\tilde{y} - \tilde{y}^*) \\ &\quad - \|C\tilde{y} - d\|^2 - \tilde{y}^T L\tilde{y} \\ &\leq -2(\tilde{y} - \tilde{y}^*)^T Q (\tilde{y} - \tilde{y}^*) \\ &\quad - \|C\tilde{y} - d\|^2 - \tilde{y}^T L\tilde{y}. \end{aligned} \quad (14)$$

Furthermore, since $C\tilde{y}^* = d$ and $L\tilde{y}^* = 0$, it results in

$$\begin{aligned} &\dot{V}(\tilde{y}(t), \tilde{v}(t), \tilde{w}(t)) \\ &\leq -(\tilde{y} - \tilde{y}^*)^T (2Q + C^T C + L) (\tilde{y} - \tilde{y}^*). \end{aligned} \quad (15)$$

From the LaSalle invariance principle [33], $\tilde{y}(t)$ is convergent to a set E as $t \rightarrow \infty$, where E is the largest invariant set of

$$\Theta = \{\tilde{y} \in \mathbb{R}^{mn} : \dot{V}(\tilde{y}(t), \tilde{v}(t), \tilde{w}(t)) = 0\}.$$

Since the matrix $2Q + C^T C + L$ is positive definite, the set Θ has a unique element \tilde{y}^* . Then \tilde{y} is

globally convergent to the unique optimal solution \tilde{y}^* ; i.e., the state vectors $y^i(t)$ of the system reach consensus at the unique optimal solution of (1) if $2Q + C^T C + L$ is positive definite and the graph of the multi-agent network is connected and undirected.

3.3 Consensus Analysis for Positive Semi-definite Q

If the objective function in (1) is convex, the KKT condition in (3) is also sufficient. Then the distributed algorithm is always capable of solving the quadratic programming problems with convex objective functions.

Theorem 5 *The state vectors $y^i(t)$ ($i = 1, 2, \dots, m$) of system (9) reach consensus at an optimal solution of (1) if Q is positive semi-definite and the graph of the multi-agent network is connected and undirected.*

Proof From (14), if Q is positive semi-definite, we have

$$\begin{aligned} & \dot{V}(\tilde{y}(t), \tilde{v}(t), \tilde{w}(t)) \\ & \leq -2\|\tilde{y} - \tilde{\phi}\|^2 - \|C\tilde{y} - d\|^2 - \tilde{y}^T L \tilde{y}. \end{aligned}$$

The remainder of the proof is similar to the proof of Theorem 2 in [17], so we omit it here. \square

Remark 3.1 *From the results in Theorem 4, we can see that if the matrix Q or $2Q + C^T C$ is not positive definite, the matrix $2Q + C^T C + L$ may be positive definite after coupling the decomposable nonconvex quadratic programming. Moreover, since KKT conditions are necessary for non-convex programming, the optimal solution can be guaranteed if the KKT point is unique. One of the conditions is of the positive definite matrix $2Q + C^T C + L$.*

4 Illustrative Example

Consider the decomposable quadratic programming problem (1) with

$$Q_1 = \begin{pmatrix} 2 & 0.5 & 0 \\ 0.5 & 0.5 & 0 \\ 0 & 0 & -0.5 \end{pmatrix}, Q_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0.5 \\ 0 & 0.5 & -0.5 \end{pmatrix},$$

$$Q_3 = \begin{pmatrix} 1.5 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & -0.5 \end{pmatrix},$$

$$q_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, q_2 = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}, q_3 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

$$C_1 = C_2 = C_3 = \begin{pmatrix} 1 & -1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

$$d_1 = d_2 = d_3 = \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

and

$$\Omega_1 = \Omega_2 = \Omega_3 = [-1, 1]^3.$$

Since $Q_1 + Q_2 + Q_3$ is not positive semi-definite, the quadratic programming problem is not convex. In the simulations, we use a three-agents network with a circle connected graph to solve this problem. The connection weight is 1 between any two agents. The agent i is allocated the objective function with Q_i and q_i and the constraints of all the agents are same. The output transient behaviors of the three agents on y are shown in Fig. 1, from which we can see that the state variables of the network reach a consensus at the optimal solution $y^* = (-0.5, -1, 0.5)^T$.

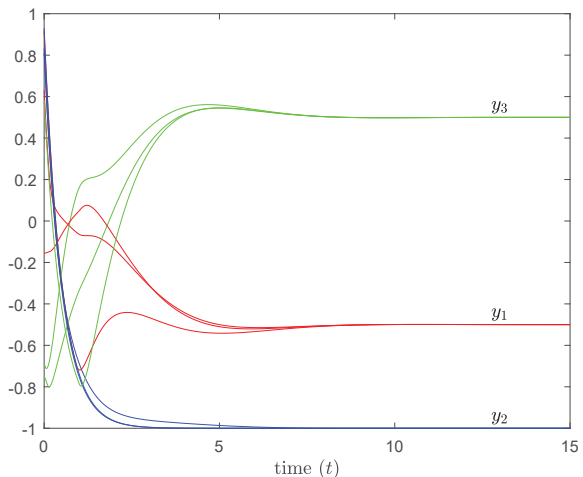


Figure 1. Consensus of the output vector y of the multi-agent network.

5 Conclusion

This paper presented a distributed algorithm for a class of decomposable nonconvex quadratic programming. The distributed algorithm was realized by the multi-agent network with connected

topology. The algorithm was proved to be convergent to optimal solution with consensus under connected graph. Compared with existing algorithms for quadratic programming, the proposed algorithm was suitable for solving more general distributed quadratic programming problems with nonconvex objective functions. Moreover, a numerical example was presented to show the algorithm's performance.

Appendix

A. Graph Theory

Generally, a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ with order m is composed of three parts: a vertex set $\mathcal{V} = \{1, 2, \dots, m\}$, an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and an adjacency matrix $\mathcal{A} = \{a_{ij}\}_{m \times m}$ with nonnegative elements $a_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}$, otherwise $a_{ij} = 0$. The graph is undirected if $a_{ij} = a_{ji}$. If the nodes i and j can exchange information with each other, we say that an undirected edge e_{ij} is in the graph, denoted by an unordered pair of nodes (i, j) . If the graph is undirected, it indicates that the communications among agents are bidirectional. It is noted that $a_{ii} = 0$ which indicates no self-connection in the graph. A graph is connected if there exists a path between any pair of distinct nodes i and j . The degree of node i is defined by $d_i = \sum_{j=1, j \neq i}^m a_{ij}$ ($i = 1, 2, \dots, m$). The Laplacian matrix of graph is defined as $L_m = D - \mathcal{A}$, in which D is diagonal with $D = \text{diag}\{d_1, d_2, \dots, d_m\}$.

B. Projection Operator

In general, a projection operator from \mathbb{R}^n to $\Omega \subseteq \mathbb{R}^n$ can be defined as

$$\phi(\alpha) = \arg \min_{\beta \in \Omega} \|\beta - \alpha\|,$$

where Ω is a closed convex set.

As two special cases, if Ω is a box set as $\Omega = \{\alpha \in \mathbb{R}^n : l_i \leq \alpha_i \leq h_i, i = 1, 2, \dots, n\}$,

$$\phi(\alpha_i) = \begin{cases} h_i, & \alpha_i > h_i, \\ u_i, & l_i \leq \alpha_i \leq h_i, \\ l_i, & \alpha_i < l_i. \end{cases}$$

Or if Ω is a sphere set $\Omega = \{\alpha \in \mathbb{R}^n : \|\alpha - s\| \leq$

$r, s \in \mathbb{R}^n, r > 0\}$,

$$\phi(\alpha) = \begin{cases} \alpha, & \|\alpha - s\| \leq r, \\ s + \frac{r(\alpha - s)}{\|\alpha - s\|}, & \|\alpha - s\| > r. \end{cases}$$

Lemma 6 [31] *The projection operator always satisfies the following inequality*

$$(\omega - \phi(\omega))^T (\phi(\omega) - \nu) \geq 0, \quad \forall \omega \in \mathbb{R}^n, \nu \in \Omega.$$

References

- [1] R. H. Byrd, M. E. Hribar, and J. Nocedal, An interior point algorithm for large-scale nonlinear programming, *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.
- [2] M. A. Figueiredo, R. D. Nowak, and S. J. Wright, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, Dec. 2007.
- [3] J. Hopfield and D. Tank, Computing with neural circuits: A model, *Science*, vol. 233, no. 4764, pp. 625–633, 1986.
- [4] Y. Xia, G. Feng, and J. Wang, A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints, *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1340–1353, Aug. 2008.
- [5] Q. Liu and J. Wang, L_1 -minimization algorithms for sparse signal reconstruction based on a projection neural network, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 698–707, Mar. 2016.
- [6] Q. Liu and J. Wang, A projection neural network for constrained quadratic minimax optimization, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2891–2900, Nov. 2015.
- [7] G. Tambouratzis, Using particle swarm optimization to accurately identify syntactic phrases in free text, *Journal of Artificial Intelligence & Soft Computing Research*, vol. 8, no. 1, pp. 63–67, 2018.
- [8] S. Sadiqbatcha, J. Saeed, and A. Yiannis, Particle swarm optimization for solving a class of type-1 and type-2 fuzzy nonlinear equations, *Journal of Artificial Intelligence & Soft Computing Research*, vol. 8, no. 2, pp. 103–110, 2018.
- [9] C. Rotar and L. B. Iantovics, Directed evolution - a new metaheuristic for optimization, *Journal of Artificial Intelligence & Soft Computing Research*, vol. 7, no. 3, pp. 183–200, 2017.

- [10] J. Antonio, G. Huang, and W. Tsai, A fast distributed shortest path algorithm for a class of hierarchically clustered data networks, *IEEE Transactions on Computers*, pp. 710–724, 1992.
- [11] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, A new class of distributed optimization algorithms: Application to regression of distributed data, *Optimization Methods and Software*, vol. 27, no. 1, pp. 71–88, 2012.
- [12] L. Xiao, S. Boyd, and S.-J. Kim, Distributed average consensus with least-mean-square deviation, *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.
- [13] A. Nedic, A. Ozdaglar, and P. A. Parrilo, Constrained consensus and optimization in multi-agent networks, *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.
- [14] M. Zhu and S. Martínez, On distributed convex optimization under inequality and equality constraints, *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, Jan. 2012.
- [15] D. Yuan, S. Xu, and H. Zhao, Distributed primal-dual subgradient method for multiagent optimization via consensus algorithms, *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, vol. 41, no. 6, pp. 1715–1724, Dec. 2011.
- [16] B. Ghahserifard and J. Cortés, Distributed continuous-time convex optimization on weight-balanced digraphs, *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, Mar. 2014.
- [17] Q. Liu and J. Wang, A second-order multi-agent network for bound-constrained distributed optimization, *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3310–3315, Dec. 2015.
- [18] M. Rabbat and R. Nowak, Distributed optimization in sensor networks, in *Proc. 3rd International Symposium on Information Processing in Sensor Networks*, Berkeley, CA, USA, Apr. 2004, pp. 20–27.
- [19] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, Distributed optimization with local domains: Applications in MPC and network flows, *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 2004–2009, July 2015.
- [20] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer-Verlag London Limited, 2008.
- [21] A. Nedić and A. Ozdaglar, Subgradient methods for saddle-point problems, *Journal of optimization theory and applications*, vol. 142, no. 1, pp. 205–228, 2009.
- [22] I. Lobel, A. Ozdaglar, and D. Feijer, Distributed multi-agent optimization with state-dependent communication, *Mathematical Programming*, vol. 129, no. 2, pp. 255–284, 2011.
- [23] P. Lin, W. Ren, and Y. Song, Distributed multi-agent optimization subject to nonidentical constraints and communication delays, *Automatica*, vol. 65, pp. 120–131, 2016.
- [24] M. Bürger, G. Notarstefano, and F. Allgöwer, A polyhedral approximation framework for convex and robust distributed optimization, *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 384–395, Feb. 2014.
- [25] L. Carlone, V. Srivastava, F. Bullo, and G. C. Calafiore, Distributed random convex programming via constraints consensus, *SIAM Journal on Control and Optimization*, vol. 52, no. 1, pp. 629–662, 2014.
- [26] X. Wang, Y. Hong, and H. Ji, Distributed optimization for a class of nonlinear multiagent systems with disturbance rejection, *IEEE Transactions on Cybernetics*, vol. 46, no. 7, pp. 1655–1666, July 2016.
- [27] S. Yang, Q. Liu, and J. Wang, Distributed optimization based on a multiagent system in the presence of communication delays, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 5, pp. 717–728, May 2017.
- [28] H. Wang, X. Liao, T. Huang, and C. Li, Cooperative distributed optimization in multiagent networks with delays, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 2, pp. 363–369, Feb. 2015.
- [29] Q. Liu, S. Yang, and J. Wang, A collective neurodynamic approach to distributed constrained optimization, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 8, pp. 1747–1758, Aug. 2017.
- [30] M. Bazaraa, H. Sherali, and C. Shetty, *Nonlinear Programming: Theory and Algorithms (3rd Ed.)*. Hoboken, New Jersey: John Wiley & Sons, 2006.
- [31] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, New York: Academic, 1982.
- [32] Q. Liu and K. Li, A continuous-time algorithm based on multi-agent system for distributed least absolute deviation subject to hybrid constraints,” in *Proc. 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 7381–7386.
- [33] J. LaSalle, *The Stability of Dynamical Systems*. Philadelphia, PA, USA: SIAM, 1976.



Yan Zhao received the B.S. and M.S. degrees in Mathematics from Anhui Normal University, Wuhu, China, in 2001 and 2012 respectively. She is currently a lecturer with the School of Common Courses, Wannan Medical College, Wuhu, China. Her current research interests include computational mathematics and optimization theory.



Qingshan Liu received the B.S. degree in Mathematics from Anhui Normal University, Wuhu, China, in 2001, the M.S. degree in applied mathematics from Southeast University, Nanjing, China, in 2005, and the Ph.D. degree in automation and computer-aided engineering, Chinese University of Hong Kong, Hong Kong, in 2008. He is currently a Professor with the School of Mathematics, Southeast University, Nanjing, China. His current research interests include optimization theory and applications, artificial neural networks, computational intelligence, and multi-agent systems. Dr. Liu serves as an Associate Editor of the *IEEE Transactions on Cybernetics* and *IEEE Transactions on Neural Networks and Learning Systems*, and is a member of the Editorial Board of *Neural Networks*.