A Decomposition/Synchronization Scheme for Formulating and Solving Optimization Problems

James Anderson * Antonis Papachristodoulou**

* Department of Engineering Science, University of Oxford, OX1 3PJ, U.K. (e-mail: james.anderson@eng.ox.ac.uk)
** Department of Engineering Science, University of Oxford, OX1 3PJ, U.K. (e-mail: antonis@eng.ox.ac.uk)

Abstract: Large-scale optimization problems, even when convex, can be challenging to solve directly. Recently, a considerable amount of research has focused on developing methods for solving such optimization problems in a distributed manner. The assumption that is usually made is that the global objective function is a sum of convex functions, which is restrictive.

In this paper, we automatically decompose a convex function to be minimized into a sum of smaller functions that may or may not be convex and assign each sub-function to an agent in a networked system. Each agent is allowed to communicate with other agents in order to solve the original optimization problem. We propose an algorithm which will converge when the interaction between the agents is strong enough to lead to synchronization between common variables.

Keywords: Distributed Optimization, Synchronization, Lyapunov Stability.

1. INTRODUCTION

Understanding how large-scale dynamical systems, also called networked systems, function is an important research topic. Typical systems of interest, to name just a few, include Internet resource allocation (Srikant, 2003), agreement, alignment and synchronization (Strogatz, 2003), multi-agent system consensus (Jadbabaie et al., 2003), flocking (Olfati-Saber, 2006), sensor networks (Boyd et al., 2006) etc. Such systems contain a large number of agents interacting on a network with a known or unknown topology. They can all be thought of as trying to optimize a global network objective for which a distributed solution is being sought. It is not usually obvious how to decompose a large optimization problem into a sum of smaller subproblems so that if these are assigned to agents they can cooperatively solve the original problem using only locally available information. Frequently it is assumed that the problem has already been decomposed into smaller problems that have certain desirable properties, such as convexity.

Methods for solving convex optimization problems in a distributed manner are derived and analyzed in Bertsekas and Tsitsiklis (1989). Recently, Nedić and Ozdaglar (2009) extended the framework of optimizing a sum of convex functions to include cases where the objective function is not necessarily smooth and may be subject to a time-varying topology. They also provide the first convergence analysis for such a system. In Nedić and Ozdaglar (2008) their convergence analysis is extended to deal with systems with communication delays on the information exchange.
illustrating the algorithm and its convergence properties before we present the general case in Section 4. In section 5 we illustrate our methodology with an example. The paper is concluded in Section 6.

2. BACKGROUND INFORMATION

In this section we present a brief review of previous results on distributed optimization, algebraic graph theory, graph partitioning and synchronization that we will use in the sequel.

2.1 Distributed Optimization

Consider the problem of minimizing a sum of convex functions in a distributed manner – assume that the optimization problem is unconstrained:

\[
\min f(x) = \sum_{i=1}^{N} f_i(x) \quad \text{s.t. } x \in \mathbb{R}^n
\]

where the functions \( f_i : \mathbb{R}^n \rightarrow \mathbb{R} \) are convex. Let the optimal solution vector to (1) be \( x^* \); the optimal value is \( f^* \). If the function \( f(x) \) is strictly convex, then \( x^* \) is a unique global minimum.

Suppose each function \( f_i \) is assigned to an agent that attempts to minimize it; if the function is separable, i.e., if each variable \( x_i \) is found in only one subfunction \( f_i \), then the problem decouples and can be solved easily. If the function is non-separable, unless there is communication between the \( N \) agents, at the end each agent will hold a local estimate of the optimal solution but the value for \( x \) will not be equal for all the agents. Agent communication is essential in order to ‘synchronize’ these variables.

In this case, each agent updates its estimate of the optimal value by combining its current estimate (own information) with information provided by its neighbours (local information). Denote by \( y_i \) the decision variables held by agent \( i \), then the original problem can be cast as follows

\[
\min \sum_{i=1}^{N} f_i(y_i) \quad \text{s.t. } y_{ik} = y_{jk}, \quad \forall \text{ agents } i,j \text{ sharing variable } x_k
\]

In Nedić and Ozdaglar (2009), it was argued that a discrete time system can be constructed which has dynamics that can be interpreted as follows: one term represents a gradient descent with a positive step-size \( \alpha_i \) based on the agent’s current estimate, while a second term incorporates information from neighboring agents weighted accordingly in order to synchronize those values.

The contribution of this paper is to describe a method for solving large unconstrained optimization problems in a distributed manner when the objective function cannot be decomposed into a sum of convex functions.

2.2 Algebraic Graph Theory

In this section we provide a brief overview of some ideas from algebraic graph theory that will be used in the sequel.

A graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) consists of a set of vertices (nodes) \( \mathcal{V} = \{v_j, j \in N = \{1, \ldots, N\} \} \), which represent the agents, and a set of edges (links) \( \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \), which represent the interaction between the agents. If \( v_i, v_j \in \mathcal{V} \) and \( e_{ij} = (v_i, v_j) \in \mathcal{E} \), then there is an edge from node \( v_i \) to node \( v_j \), i.e., agent \( j \) and \( i \) can exchange data and are neighbours. In this paper, we assume that the graph \( \mathcal{G} \) is undirected so that \( e_{ij} \in \mathcal{E} \) if and only if \( e_{ji} \in \mathcal{E} \). We also assume that the network topology does not contain self-loops so that \( e_{ii} \notin \mathcal{E} \). The unweighted adjacency matrix \( A \) defined by

\[
A_{ij} = \begin{cases} 1 & \text{if } e_{ij} \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}
\]
captures the structure of the interactions of the underlying graph.

The number of neighbours of each vertex, \( n_i \), is called the degree or valency of vertex \( v_i \). This information is used to construct the diagonal degree matrix \( D = \text{diag}(n_i) \). The Laplacian matrix of the graph is defined by \( L = D - A \), which is a positive semi-definite matrix, \( L \succeq 0 \). An important property of \( L \) is that it always has an eigenvalue of 0 associated with an eigenvector of ones, i.e., \( L \mathbf{1} = 0 \) where \( \mathbf{1} \) is the vector of ones. The second smallest eigenvalue of \( L \), known as the Fiedler eigenvalue, denoted by \( \lambda_2(L) \), will play an important role later on, when we describe methods for graph decomposition. For a comprehensive discussion on graph theory the reader is directed to Godsil and Royle (2000).

2.3 Decomposition Methods

Given a graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) with adjacency matrix \( A \), the K-way partitioning problem requires one to construct \( K \) subgraphs, \( \mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k) \), \( k = 1, \ldots, K \) so that \( \bigcup_{k=1}^{K} \mathcal{V}_k = \mathcal{V} \), \( \mathcal{V}_k \cap \mathcal{V}_\ell = \emptyset \) for all \( k, \ell = 1, \ldots, K \), and \( \mathcal{E}_k = \{(v_i, v_j) \in \mathcal{E} | v_i \in \mathcal{V}_k, v_j \in \mathcal{V}_k \} \) where the number of nodes in each subgraph is approximately equal. This partition problem is NP-complete.

In this paper we are not concerned with creating equally sized partitions, but instead are interested in weighted graph decomposition, where the weights denote the strength of the interactions. In other words, our objective is to find a partition that minimizes the sum of the weights on the edges linking the subgraphs. This may be interpreted as trying to minimize the communication between nodes in different subgraphs.

The above problem can be formulated as follows. Given a symmetric graph with a weighted adjacency matrix \( A \), we want to assign integer values \( z_i = \pm 1, i = 1, \ldots, N \) to the graph nodes so as to minimize

\[
h(z) = \frac{1}{4} \sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} (z_i - z_j)^2
\]

while excluding the case in which all the \( z_i \)'s have equal sign (Karisch et al., 2000). The reasoning is the following: if \( z_i \) and \( z_j \) are neighbours and have the same sign, they belong to the same subgraph and hence do not contribute to the function \( h(z) \), otherwise this function adds all the weights of edges that connect nodes from one subgraph to the other. In short, and using the weighted graph Laplacian \( L = D - A \) where \( D = \text{diag}(AA^T) \) we have:
\[
\min \frac{1}{4} z^T L z \\
\text{s.t. } z_i^2 = 1, \\
z^T 1 \neq \pm N.
\]

If we drop the two constraints, then the function \( h(z) \) is minimized (apart from the trivial case for which \( z = c1 \), where \( c \) is a constant) for \( z \) aligned with the eigenvector \( \tilde{\alpha} \) corresponding to \( \lambda_2(L) \); this is known as spectral partitioning and is computationally efficient to compute (Karypis and Kumar, 1998).

The spectral partitioning algorithm works as follows: First the eigenvector corresponding to the second eigenvalue is calculated. This eigenvector is then sorted into ascending order and the median value is computed. The corresponding sorted nodes which lie above the median value are assigned to one partition i.e., \( z_i = 1 \), and those which fall below the median assigned to the second partition \( z_i = -1 \). This method can be recursively applied until the desired number of partitions is obtained.

Other, usually more accurate, ways to compute the partitions rely on semidefinite programming relaxations, see for example Goemans and Williamson (1995).

### 2.4 Synchronization

The last piece of the puzzle is synchronization on undirected graphs. Results from synchronization are needed in order to analyze the convergence properties of the dynamical system that will try to solve the large-scale optimization problem, especially in the case of non-convex sub-problems. Here we present some results about the synchronization properties of interacting systems on undirected graphs, see, e.g., Pecora and Barahona (2005); Li and Chen (2003) for more details.

Consider a coupled system of the form:

\[
\dot{x}_i = \hat{f}(x_i) - \sigma \sum_{j=1}^{N} L_{ij} G(x_j)
\]

where \( x_i \in \mathbb{R}^n \) is the state vector, \( \hat{f}(x_i) \) describes the state dynamics of the (identical) oscillators and \( \sigma \) is the coupling strength between the agents. Also, \( L \) is the (symmetric) Laplacian matrix, which encodes which agents are allowed to communicate and the function \( G : \mathbb{R}^n \to \mathbb{R}^n \) is an output function that determines which states of the agents are being communicated to other agents – the simplest case is when \( G \in \mathbb{R}^{n \times n} \), which we assume now for simplicity.

We are interested in the synchronization properties of the above system, i.e., whether \( x_1 = x_2 = \ldots = x_n = x^* \) asymptotically. For this to happen, we need to assume that \( \hat{f}(x^*) = 0 \). Stability of the synchronized state can then be tested by introducing small perturbations to the steady state and observing if they die out or grow over time.

Introducing small perturbations \( \zeta_i \) into the system at steady state such that \( x_i(t) = x_i^* + \zeta_i(t) \) and carrying out a Taylor expansion results in the following perturbation dynamics:

\[
\dot{\zeta}_i = \sum_{j=1}^{N} \left[ D\hat{f}(x^*) \delta_{ij} - \sigma L_{ij} G \right] \zeta_j
\]

where \( D\hat{f}(x^*) \) denotes the Jacobian matrix of \( \hat{f}(x_i) \) evaluated at \( x^* \) and \( \delta_{ij} \) is the Kronecker delta (for which \( \delta_{ij} = 1 \) if \( i = j \) and \( \delta_{ij} = 0 \) otherwise). The block structure of the above equations can be exploited in order to analyze the stability of the synchronized state. In particular, we get

\[
\dot{\zeta} = I_N \otimes D\hat{f}(x^*)(\zeta - \sigma(L \otimes G)\zeta)
\]

where \( \otimes \) denotes the Kronecker product \(^1\) and \( I_N \) is the \( N \times N \) identity matrix. Now let us diagonalize \( L = V A V^T \) with \( V \) orthonormal and \( A \) diagonal with non-negative entries. Since \( L \otimes G = (V A V^T) \otimes G = (V \otimes I_n)(A \otimes G)(V^T \otimes I_n) \), we get, after a change of coordinates \( \xi = (V^T \otimes I_n)\zeta \):

\[
\dot{\xi} = I_N \otimes D\hat{f}(x^*)(\xi - \sigma(A \otimes G)\xi)
\]

or, for each agent,

\[
\dot{\xi}_i = D\hat{f}(x^*)(\xi_i - \sigma(G)\xi_i)
\]

where \( \sigma_i = \sigma \lambda_i(L) \). We have now diagonalized the dynamics and the stability analysis is simplified. For particular cases, it is easy to develop conditions on \( \sigma \) for synchronization – see (Pecora and Carroll, 1997) for more details. This result assumes that both the \( \hat{f}(x_i) \) and the coupling matrices \( G \) have to be identical for all agents and interactions respectively.

### 3. MOTIVATING EXAMPLE

We now discuss a simple example to show how the results in the previous section will be combined and used to solve the problem of distributed optimization efficiently, even if the original optimization problem cannot be separated into convex sub-problems. Consider an unconstrained convex optimization problem in three variables:

\[
\min \ f(x_1, x_2, x_3)
\]

where \( f \) is a convex function which can be separated in the following way:

\[
f(x_1, x_2, x_3) = f_1(x_1, x_2) + f_2(x_2, x_3)
\]

Here, the functions \( f_1 \) and \( f_2 \) are not necessarily convex. Our aim is to assign the two functions to two computers each of which will solve an appropriate minimization problem, so that the original problem is also solved, even approximately. For this to happen, however, the values of \( x_2 \) that both of the subsystems hold need to be ‘synchronized’.

We assume that subproblem 1 has a state \( y_1 = [y_{11}, y_{12}, y_{13}] \) and subproblem 2 has a state \( y_2 = [y_{21}, y_{22}, y_{23}] \), even if some of these state elements do not need to be updated. This means that the original problem can be written as:

\[
\min \ f_1(y_{11}, y_{12}) + f_2(y_{22}, y_{23})
\]

s.t. \( y_{12} = y_{22} = 0 \).

The equality constraint represents the information exchange between the two subproblems.

Consider the following augmented Lagrangian function

\[
V(y_{11}, y_{12}, y_{22}, y_{23}) = f_1(y_{11}, y_{12}) + f_2(y_{22}, y_{23}) + \gamma \int_{0}^{y_{12} - y_{22}} g(z) dz
\]

\(^1\) The Kronecker product of matrices \([A_{ij}]\) and \( B \), denoted by \( A \otimes B \), is equal to \([A_{ij}B]\).
where $\gamma > 0$ and $g(z)$ is a non-decreasing, continuous function that acts as a ‘barrier’ and satisfies:

$$
g(0) = 0, \quad \int_0^\infty g(z)dz \to \infty \text{ as } \|z\| \to \infty.
$$

A simple condition for this to happen is for $g(z)$ to be strictly globally passive 2. In combination with a gradient descent algorithm on the above Lagrangian, we obtain the following dynamics that the agents need to use in order to update their estimates of the optimal solution:

$$
\dot{y}_{11} = -\alpha_{11} \frac{\partial f_1}{\partial y_{11}} \\
\dot{y}_{12} = -\alpha_{12} \left( \frac{\partial f_1}{\partial y_{12}} + \gamma(g(y_{11} - y_{22})) \right) (7) \\
\dot{y}_{22} = -\alpha_{22} \left( \frac{\partial f_2}{\partial y_{22}} + \gamma(g(y_{22} - y_{11})) \right) \\
\dot{y}_{23} = -\alpha_{23} \frac{\partial f_2}{\partial y_{23}}
$$

The equilibrium point of (7) satisfies:

$$
\frac{\partial f_1}{\partial y_{11}} = 0 \\
\frac{\partial f_1}{\partial y_{12}} + \gamma(g(y_{11} - y_{22})) = 0 \\
\frac{\partial f_2}{\partial y_{22}} + \gamma(g(y_{22} - y_{11})) = 0 \\
\frac{\partial f_2}{\partial y_{23}} = 0
$$

The first order optimality conditions for the original optimization problem (4) are:

$$
\frac{\partial f_1}{\partial x_1} = 0 \\
\frac{\partial f_1}{\partial x_2} + \frac{\partial f_2}{\partial x_2} = 0 \\
\frac{\partial f_2}{\partial x_3} = 0
$$

when evaluated about the (unique) optimal point $(x_1^*, x_2^*, x_3^*)$. Obviously, when $y_{22} = y_{11}$ we obtain the same equilibrium point as the optimal solution to (4), with the additional constraint $\frac{\partial f_2}{\partial y_{22}} = \frac{\partial f_1}{\partial y_{12}}$ at equilibrium. Hence this equilibrium point is a sub-optimal solution to the original optimization problem.

In the case in which $f_i$ are convex functions, the augmented Lagrangian (6) is a Lyapunov function (Khalil, 2001) of system (7), i.e., it is locally positive definite and its time-derivative is negative semi-definite. This is so, as it is the aggregate sum of convex functions $f_1(y_{11}, y_{12})$ and $f_2(y_{22}, y_{23})$ and the second term, $g(0) = 0$, is positive definite by construction, as $g$ is strictly passive – independent of the value of $\gamma > 0$. The derivative of (6) around the synchronized state is

$$
\dot{V} = -\sum_{i=1}^N \sum_{j=1}^n \frac{1}{\alpha_{ij}} \dot{y}_{ij}^2
$$

A simple LaSalle argument verifies that the equilibrium is asymptotically attracting. Therefore the above dynamics will converge to the synchronized state, which is the unique minimum of the Lyapunov function shown above. As already mentioned, this method will converge for all $\gamma$ since the subfunctions $f_i(x_i)$ are convex. This conclusion generalizes to more complex cases, involving several convex sub-functions in many variables.

The case of non-convex sub-functions $f_i$ is more interesting and will be examined now. In fact, it is unlikely that the subproblems will be convex as the method we will be using to construct the subfunctions will not guarantee this property. In this case, the function (6) may or may not be locally positive definite, and that will depend on the value of $\gamma$, the coupling strength between the agents. In this case, the Hessian of the Lyapunov function about this equilibrium is of the form:

$$
\begin{bmatrix}
\frac{\partial^2 f_1}{\partial y_{11}^2} & \frac{\partial^2 f_1}{\partial y_{11} \partial y_{12}} & \frac{\partial^2 f_1}{\partial y_{12}^2} & \frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{22}^2} & \frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{23}^2} \\
\frac{\partial^2 f_1}{\partial y_{11} \partial y_{12}} & \frac{\partial^2 f_1}{\partial y_{12}^2} & \frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{22}^2} & \frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{23}^2} \\
\frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{12}^2} & \frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{22}^2} & \frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{23}^2} \\
\frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{12} \partial y_{22}} & \frac{\partial^2 f_1}{\partial y_{22}^2} & \frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{23}^2} \\
\frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{22}^2} & \frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{22} \partial y_{23}} & \frac{\partial^2 f_1}{\partial y_{23}^2} \\
\frac{\partial^2 f_1}{\partial y_{23}^2} & \frac{\partial^2 f_1}{\partial y_{23}^2} & \frac{\partial^2 f_1}{\partial y_{23}^2} & \frac{\partial^2 f_1}{\partial y_{23}^2} & \frac{\partial^2 f_1}{\partial y_{23}^2} & \frac{\partial^2 f_1}{\partial y_{23}^2} & \frac{\partial^2 f_1}{\partial y_{23}^2} \\
\end{bmatrix}
$$

This can be made positive definite by choice of $\gamma$ (as we will show in the next section) in order to ‘synchronize’ the agents’ estimates of the global optimal value. Note that the linearization of the dynamics of system (7) about the equilibrium has the same structure as the above Hessian matrix. In that case, at least locally, the synchronized equilibrium is asymptotically attracting.

4. RESULTS

Let us now consider the unconstrained optimization problem

$$
\min f(x)
$$

where $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and convex. Here $n$ is large and we assume that $f(x)$ is the aggregate sum of several terms (as in (1)). Since the function $f(x)$ is convex, the minimum can be found by solving the set of simultaneous equations

$$
\frac{\partial f}{\partial x} = 0.
$$

We want to represent the ‘interaction’ between the variables as they appear in the terms of the above function using a graph. For this purpose, we construct a weighted graph $G(V, E)$ where the number of nodes is equal to the number of terms in $f$; an edge between two nodes indicates that they share the same variable. The relevant matrices described in Section (2.3) are then constructed for $G$ and used to partition the graph into a number of smaller graphs which define a set of sub-functions $f_i(x)$, such that $f(x) = \sum_{i=1}^N f_i(x)$, which have a small number of common variables. Next we define:

- $A \in \mathbb{R}^{N \times N}$ the adjacency matrix between the agents, whose $(i,j)$th entry is 1 if $f_i(x)$ and $f_j(x)$ share a variable and is 0 otherwise.
- $G^{ij} \in \mathbb{R}^{n \times n}$, is 0 – 1 diagonal coupling matrix, whose $(k, k)$ entry indicates that the $k^{th}$ state of agent $j$ is needed by agent $i$.

\[ A \in \mathbb{R}^{N \times N} \text{ the adjacency matrix between the agents, whose (i, j)th entry is 1 if f_i(x) and f_j(x) share a variable and is 0 otherwise.} \]

\[ G^{ij} \in \mathbb{R}^{n \times n}, \text{ is 0 – 1 diagonal coupling matrix, whose (k, k) entry indicates that the k^{th} state of agent j is needed by agent i.} \]
Note that these matrices describe the agent and state interactions, and not the interaction between the terms in the original problem described by \( G \). We now assign each function \( f_i(x) \) to each of the \( N \) agents and denote the state that each agent holds as \( y_i = [y_{ij}] \) for \( j = 1, \ldots, n \) and \( i = 1, \ldots, N \). Each agent will try to minimize the function \( f_i(y_i) \), under the constraint that \( y_{ik} = y_{jk} \) for all states \( k \) that are shared by agents \( i \) and \( j \):

\[
\min_{\mathbf{y}} \sum_{i=1}^{N} f_i(y_i) \\
\text{s.t. } y_{ik} = y_{jk} \text{ for all } A_{ij} \neq 0 \text{ and } G_{kk} = 1.
\]

Clearly, this optimization problem is the same as the original one.

Considering an augmented Lagrangian function for the original problem, we obtain

\[
L(y, \gamma) = \sum_{i=1}^{N} f_i(y_i) \\
+ \frac{\gamma}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{n} A_{ij} G_{kk} g(y_{ik} - y_{jk})
\]

where we have assumed identical coupling functions that are strictly passive and \( \gamma > 0 \). The gradient dynamics in this case (ignoring zero dynamics) become:

\[
\dot{y}_{ik} = -\alpha_{ik} \left( \frac{\partial f_i(y_i)}{\partial y_{ik}} + \gamma \sum_{j=1}^{N} A_{ij} G_{kk} g(y_{ik} - y_{jk}) \right)
\]

where the sub-gradients are parameterized by \( \alpha_{ik} > 0 \) and the interactions by the coupling strength parameter \( \gamma \). Note that the dynamics will be zero if \( f_i(y_i) \) is not a function of some variables; in which case, the corresponding state is not iterated at all by the agent. The convergence of these dynamics will depend on the value of \( \gamma \) which determines if the system can synchronize. If the subfunctions are all convex, then the augmented Lagrangian can serve as a Lyapunov function for the whole system:

**Theorem 1.** The synchronized equilibrium \( y_i = y^* \) for all \( i = 1, \ldots, N \) of system (9) with convex \( f_i(y_i) \) is asymptotically attracting.

**Proof.** First consider the case in which the interaction graph between the agents is connected. Consider (8) as a Lyapunov function around the synchronized equilibrium \( y_i = y^* \) for all \( i = 1, \ldots, N \) for system (9). This function is positive definite by construction, for all \( \gamma > 0 \), as it is the sum of convex functions and positive definite functions since \( g(z) \) is strictly globally passive. The time derivative of \( L(y, \gamma) \) is given by:

\[
\frac{dL}{dt} = \sum_{i=1}^{N} \sum_{k=1}^{n} \frac{\partial L}{\partial y_{ik}} \dot{y}_{ik} = -\sum_{i=1}^{N} \sum_{k=1}^{n} \frac{1}{\alpha_{ik}} \dot{y}_{ik}^2.
\]

Hence \( \frac{dL}{dt} \leq 0 \). Using a LaSalle argument, the trajectories of the system will converge to the maximal invariant set contained in the set \( \{\dot{y}_{ik} = 0\} \). Hence the synchronized equilibrium is asymptotically attracting. If the graph is not connected, the agents in each connected component operate on different variables by construction; their agreement subsets are asymptotically attracting, making the synchronized equilibrium of (9) also asymptotically attracting.

We now turn to the problem of synchronization in the case of non-convex subfunctions.

**Proposition 2.** Suppose there exists a coupling strength \( \gamma > 0 \) such that \( (H + \gamma B) \succeq 0 \), where \( H = \text{diag}(D^2 f_1, \ldots, D^2 f_N) \). The differentiation operator and the \((i,j)\) block of \( B \) is equal to \( B_{ij} = L_{ij} \otimes G^{ij} \). Then the synchronized equilibrium \( y_i = y^* \) for all \( i = 1, \ldots, N \) of system (9) is asymptotically attracting.

**Proof.** In order for the system described by (9) to be asymptotically attracting it must be shown that the augmented Lagrangian equation (8) is also a Lyapunov function for the system around the equilibrium point. This is achieved by showing that the Hessian of the Lyapunov function is positive definite about the synchronized state (at least locally).

First we linearize (8) around the equilibrium point and construct the corresponding Hessian matrix given by \( (H + \gamma B) \). The Hessian has the same block structure as that of the example shown in Section 3. Note that \( B \) defines the coupling between states, is symmetric, positive semidefinite, with zero row and column sums. The matrix \( H \) is indefinite, block diagonal. The task of proving that the system is asymptotically attracting can now be reduced to showing that there exists a \( \gamma \) such that \( H + \gamma B \succeq 0 \). This problem can then be cast as an LMI feasibility problem.

The feasibility of the LMI ensures that the Lyapunov function given by (8), defines (at least locally around the equilibrium) invariant regions, and use of a LaSalle argument, in a similar way as above, allows us to show that the synchronized equilibrium is asymptotically attracting.

What may be true, but is still unproven, is that there always exists such a \( \gamma \), which is a topic of future research. Let us now now give an example to illustrate the results presented in this paper.

5. **EXAMPLE**

This example demonstrates how to decompose a function into a series of smaller sub-functions by employing the spectral partitioning methods described earlier in the paper. The optimization problem is then solved in a distributed manner according to the methodology of the previous section. We wish to minimize the convex polynomial in four variables given below which has a global minimum at \( z^* = [0 \ 0 \ 0 \ 0]^T \) where \( f^* = 0 \):

\[
f(x) = 20x_1^2 + 1.8x_2^2x_4^2 - 0.0022x_1x_4 + 0.0012x_1x_3 + 4.3x_2^2x_4^2 \\
- 0.0057x_1x_2 + 1.3x_1^2 + 0.15x_1^2x_4^2 - 1.12x_2^2x_4 - 0.22x_1x_3^2 \\
+ 0.8x_1^2 - 2.3x_2^2x_3 + 19x_2^2 + 20x_2^2 + 0.014x_1x_2 + 4.2x_3^2 \\
- 0.012x_1x_3 + 0.0023x_2x_4 + 0.46x_2^2x_2 + 19x_3^2 + 0.8x_2^2x_3.
\]

It is clear that this function has a high degree of interaction between decision variables and there are no obvious points at which to break the function. Using the spectral partitioning method described in Section 2.3 the graph is decomposed into 3 partitions providing the following decomposition:
effect of communication delays will be subject to future research.

REFERENCES

\begin{align*}
f_1(x) &= 1.8x_2^2x_4^2 + 0.012x_4x_3 + 4.3x_2^2x_4 + 0.15x_1^2x_4x_3 + 0.8x_1^2 - 2.3x_2x_3^2x_3 + 19x_3^2 + 20x_4^2 + 0.014x_3x_2 + 4.2x_3^2 + 0.0023x_4x_4 + 0.46x_2^2x_2 + 19x_3^2 + 0.8x_4^2x_3 \\
f_2(x) &= -0.002x_1x_4 - 0.0057x_1x_2 - 1.1x_1^2x_2x_4 - 0.22x_1x_4^2 - 0.012x_1x_3 \\
f_3(x) &= 20x_2^2 + 1.3x_3^3
\end{align*}

In this case \( f_2(x) \) is a non-convex function. The graph that represents the interacting monomials is shown in Figure 1, the coloring of the nodes reflects which sub-function the monomial belongs to. The majority of the interconnections occur between monomials in the same partition, this indicates that there is a minimum of communication between the three agents.

Fig. 1. The graph structure representing the interconnection of the monomials in function \( f(x) \). Nodes colored in red, yellow and black belong to the agents that hold sub-functions \( f_1, f_2 \) and \( f_3 \) respectively.

The appropriate dynamical systems are then constructed, using a gradient-descent step size, \( \alpha = 0.5 \). A coupling strength of \( \gamma = 0.1 \) is first chosen and the system is simulated from a random initial condition. In this case the coupling strength is not enough to achieve synchronization and the system fails to converge. However, running the system again from the same initial conditions but with an increased coupling strength, \( \gamma = 0.9 \), allows the system to synchronize and convergence is achieved. In this case the dynamical system converges to the optimal point of the original optimization problem.

6. CONCLUSION

In this paper we have shown that a decomposition approach may be used to partition a large, convex optimization problem into subproblems that can be solved in a distributed fashion, even if the subproblems are not convex, by appropriate tuning of the coupling strength between the agents that operate on the sub-functions. The above method will find a suboptimal value for the optimum. An alternative algorithm would be a dual one, which will be the focus of future research.

The decomposition that we used in this paper ensures that the interaction between agents is kept to a minimum. The result of this is that the full system of equations (9) does not need to be solved as not all agents require information about all states. The rate of convergence of this, however, will depend on the strength of the interactions between states that are shared. This, and the