Convergence of the partition-based ADMM for a separable quadratic cost function

Saverio Bolognani, Ruggero Carli and Marco Todescato

I. Problem setup

Consider a network with set of nodes $V = \{1, \dots, s\}$ and fixed undirected communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $\mathcal{N}_i$ denote the set of neighbors of node i, that is, $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}$. The graph $\mathcal{G}$ is assumed to be connected. Consider the minimization of a separable cost function

$$\min_x \sum_{i=1}^{s} J_i(x) \tag{1}$$

where each $J_i : \mathbb{R}^N \to \mathbb{R}$ is a strictly convex function and it is known only to node i.

We make the following assumption.

Assumption 1. *There exists a unique solution x^* to the problem in* (1).

In this section we consider problems as in (1) with a specific structure, that is a *partition-based structure*, that we next describe. Let the vector x be partitioned as

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

where, for $i \in \{1, \dots, s\}$, $x_i \in \mathbb{R}^{m_i}$ for some $m_i \in \mathbb{N}$ such that $\sum_{i=1}^{s} m_i = N$[1]. The sub-vector x_i represents the relevant information at node i, referred to, hereafter, as the state of node i. Additionally, let us assume that the local objective functions have the same sparsity as the communication graph, namely, for $i \in \{1, \dots, s\}$, the function J_i depend only on the state of node i and on its neighbors, that is, on $\{x_j,\, j \in \mathcal{N}_i \cup \{i\}\}$. Then the problem we aim at solving distributively is

$$\min_x \sum_{i=1}^{s} J_i(x_i, \{x_j\}_{j\in\mathcal{N}_i}) \tag{2}$$

where the notation $J_i(x_i, \{x_j\}_{j\in\mathcal{N}_i})$ means that $J_i : \mathbb{R}^N \to \mathbb{R}$ is in fact a function of x_i and x_j, $j \in \mathcal{N}_i$.

To solve (2), in the next subsection we propose an iterative algorithm with the following two features

- it can be implemented in a distributed way, namely, each node needs to communicate only with its neighbors; and
- it has a partition-based structure, namely, each node keeps in memory only a copy of its own state and copies of the states of its neighbors.

S. Bolognani is with the Laboratory on Information and Decision Systems, Massachusetts Institute of Technology, Cambridge (MA), USA email: saverio@mit.edu.

R. Carli and M. Todescato are with Department of Information Engineering, University of Padova, Padova (PD), Italy email {carlirug | todescat}@dei.unipd.it

[1] According to the above partition-based structure also the optimal solution x^* is partitioned as $x^* = \left[(x_1^*)^T, \dots, (x_s^*)^T\right]^T$

In the sequel, with the notation $x_j^{(i)}$ we denote the copy of state x_j stored in memory by node i.

Motivated by real applications where the optimization problems can be cast as linear least square estimation problems, in the sequel we restrict our attention to the case where the functions J_i have the following specific quadratic form,

$$J_i(x_i, \{x_j\}_{j\in\mathcal{N}_i}) = \tag{3}$$
$$\left(z_i - A_{ii}x_i - \sum_{j\in\mathcal{N}_i} A_{ij}x_j\right)^T Q_i \left(z_i - A_{ii}x_i - \sum_{j\in\mathcal{N}_i} A_{ij}x_j\right)$$

where $z_i \in \mathbb{R}^{r_i \times m_i}$, $A_{ii} \in \mathbb{R}^{r_i \times m_i}$, $A_{ij} \in \mathbb{R}^{r_i \times m_j}$ (for $j \in \mathcal{N}_i$), and $Q_i \in \mathbb{R}^{r_i \times r_i}$, $Q_i > 0$ are given.

II. A partition-based ADMM algorithm

The method we propose in this subsection is a partition-based version of the classical ADMM method which exploits the equivalence between problem in (2) and the following problem

$$\begin{aligned}
\min_{x_i^{(i)}, \{x_j^{(i)}\}_{j\in\mathcal{N}_i},\, i\in V} \quad & \sum_{i=1}^{s} J_i(x_i^{(i)}, \{x_j^{(i)}\}_{j\in\mathcal{N}_i}) \\
\text{subject to} \quad & x_i^{(i)} = z_i^{(i,j)};\ x_j^{(i)} = z_j^{(i,j)} \\
& x_i^{(i)} = z_i^{(j,i)};\ x_j^{(i)} = z_j^{(j,i)}, \qquad \forall\, j \in \mathcal{N}_i.
\end{aligned} \tag{4}$$

Observe that the connectedness of the graph $\mathcal{G}$ and the presence of the bridge variables $z's$ ensures that the optimal solution of (4) is given by $x_i^{(i)} = x_i^*$ and $x_j^{(i)} = x_j^*$.

The redundant constraints added in problem (4) with the respect to problem (2), allow to find the optimal solution through a distributed, iterative, partition-based implementation that we next describe.

For $\rho > 0$, let the augmented Lagrangian be defined as

$$\begin{aligned}
\mathcal{L} = \sum_{i=1}^{s} \Bigg\{ & J_i(x_i^{(i)}, \{x_j^{(i)}\}_{j\in\mathcal{N}_i}) + \sum_{j\in\mathcal{N}_i} \left[\lambda_i^{(i,j)}\left(x_i^{(i)} - z_i^{(i,j)}\right)\right. \\
& \left. + \lambda_j^{(i,j)}\left(x_j^{(i)} - z_j^{(i,j)}\right)\right] + \sum_{j\in\mathcal{N}_i} \left[\mu_i^{(i,j)}\left(x_i^{(i)} - z_i^{(j,i)}\right)\right. \\
& \left. + \mu_j^{(i,j)}\left(x_j^{(i)} - z_j^{(j,i)}\right)\right] + \frac{\rho}{2} \sum_{j\in\mathcal{N}_i} \left[\|x_i^{(i)} - z_i^{(i,j)}\|^2\right. \\
& \left. + \|x_j^{(i)} - z_j^{(i,j)}\|^2 + \|x_i^{(i)} - z_i^{(j,i)}\|^2 + \|x_j^{(i)} - z_j^{(j,i)}\|^2\right] \Bigg\}
\end{aligned}$$

In our setup, we have that node i stores in memory and updates the following four vectors which contain only local information

$$X^{(i)} = \begin{bmatrix} x_i^{(i)} \\ \left\{x_j^{(i)}\right\}_{j\in\mathcal{N}_i} \end{bmatrix}; \qquad Z^{(i)} = \begin{bmatrix} \left\{z_i^{(i,j)}\right\}_{j\in\mathcal{N}_i} \\ \left\{z_j^{(i,j)}\right\}_{j\in\mathcal{N}_i} \end{bmatrix};$$

$$\Lambda^{(i)} = \begin{bmatrix} \left\{\left(\lambda_i^{(i,j)}\right)^T\right\}_{j\in\mathcal{N}_i} \\ \left\{\left(\lambda_j^{(i,j)}\right)^T\right\}_{j\in\mathcal{N}_i} \end{bmatrix},$$

and

$$\mathcal{M}^{(i)} = \begin{bmatrix} \left\{\left(\mu_i^{(i,j)}\right)^T\right\}_{j\in\mathcal{N}_i} \\ \left\{\left(\mu_j^{(i,j)}\right)^T\right\}_{j\in\mathcal{N}_i} \end{bmatrix}.$$

Let t denote the iteration index, then the ADMM cycles through three steps:

(i) Dual ascent step on the $\Lambda's$ and $\mathcal{M}'s$ variables: Node i updates the variables $\Lambda^{(i)}$ and $\mathcal{M}^{(i)}$ through a gradient ascent of $\mathcal{L}$ with step size ρ; precisely,

$$\begin{aligned}
\lambda_i^{(i,j)}(t+1) &= \lambda_i^{(i,j)}(t) + \rho\left(x_i^{(i)}(t) - z_i^{(i,j)}(t)\right) \\
\lambda_j^{(i,j)}(t+1) &= \lambda_j^{(i,j)}(t) + \rho\left(x_j^{(i)}(t) - z_j^{(i,j)}(t)\right) \\
\mu_i^{(i,j)}(t+1) &= \mu_i^{(i,j)}(t) + \rho\left(x_i^{(i)}(t) - z_i^{(j,i)}(t)\right) \\
\mu_j^{(i,j)}(t+1) &= \mu_j^{(i,j)}(t) + \rho\left(x_j^{(i)}(t) - z_j^{(j,i)}(t)\right)
\end{aligned}$$

(ii) Update of $X's$ variables: Node i updates the variable $X^{(i)}$ minimizing the augmented Lagrangian while keeping all the other variables fixed, namely,

$$\begin{aligned}
X^{(i)}(t+1) = \operatorname*{argmin}_{X^{(i)}} \Big\{ & J_i\left(x_i^{(i)}, \{x_j^{(i)}\}_{j\in\mathcal{N}_i}\right) \\
&+ \sum_{j\in\mathcal{N}_i}\Big[\lambda_i^{(i,j)}(t+1)\left(x_i^{(i)} - z_i^{(i,j)}(t)\right) \\
&\qquad + \lambda_j^{(i,j)}(t+1)\left(x_j^{(i)} - z_j^{(i,j)}(t)\right)\Big] \\
&+ \sum_{j\in\mathcal{N}_i}\Big[\mu_i^{(i,j)}(t+1)\left(x_i^{(i)} - z_i^{(j,i)}(t)\right) \\
&\qquad + \mu_j^{(i,j)}(t+1)\left(x_j^{(i)} - z_j^{(j,i)}(t)\right)\Big] \\
&+ \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\Big[\|x_i^{(i)} - z_i^{(i,j)}(t)\|^2 + \|x_j^{(i)} - z_j^{(i,j)}(t)\|^2 \\
&\qquad + \|x_i^{(i)} - z_i^{(j,i)}(t)\|^2 + \|x_j^{(i)} - z_j^{(j,i)}(t)\|^2\Big]\Big\}
\end{aligned}$$

(iii) Update of $Z's$ variables: Node i updates the variable $Z^{(i)}$ minimizing the augmented Lagrangian while keeping all the other variables fixed, namely,

$$\begin{aligned}
&Z^{(i)}(t+1) = \\
&\operatorname*{argmin}_{Z^{(i)}} \Big\{ \sum_{j\in\mathcal{N}_i}\Big[\lambda_i^{(i,j)}(t+1)\left(x_i^{(i)}(t+1) - z_i^{(i,j)}\right) \\
&\qquad + \lambda_j^{(i,j)}(t+1)\left(x_j^{(i)}(t+1) - z_j^{(i,j)}\right)\Big] \\
&\quad + \sum_{j\in\mathcal{N}_i}\Big[\mu_j^{(j,i)}(t+1)\left(x_j^{(j)}(t+1) - z_j^{(i,j)}\right) \\
&\qquad + \mu_i^{(j,i)}(t+1)\left(x_i^{(j)}(t+1) - z_i^{(i,j)}\right)\Big] \\
&\quad + \frac{\rho}{2}\sum_{j\in\mathcal{N}_i}\Big[\|x_i^{(i)}(t+1) - z_i^{(i,j)}\|^2 + \|x_j^{(i)}(t+1) - z_j^{(i,j)}\|^2 \\
&\qquad + \|x_j^{(j)}(t+1) - z_j^{(i,j)}\|^2 + \|x_i^{(j)}(t+1) - z_i^{(i,j)}\|^2\Big]\Big\}
\end{aligned}$$

Proposition 1. *Consider the partition-based ADMM algorithm described above. Let ρ be any real number. Then the trajectory $t \to \left\{X^{(i)}(t)\right\}$ converge exponentially to the optimal solution, namely, for $i \in \{1,\dots,n\}$, $x_j^{(i)}(t) \to x_j^*$ for all $j \in \mathcal{N}_i$ and, in particular,*

$$x_i^{(i)}(t) \to x_i^*.$$

Proof: Let X, Z, Λ and $\mathcal{M}$ be the vectors obtained by stacking together the vectors $\left\{X^{(i)}\right\}_{i\in V}$, $\left\{Z^{(i)}\right\}_{i\in V}$, $\left\{\Lambda^{(i)}\right\}_{i\in V}$ and $\left\{\mathcal{M}^{(i)}\right\}_{i\in V}$, respectively, namely,

$$X = \begin{bmatrix} X^{(1)} \\ X^{(2)} \\ \vdots \\ X^{(s)} \end{bmatrix}, \qquad Z = \begin{bmatrix} Z^{(1)} \\ Z^{(2)} \\ \vdots \\ Z^{(s)} \end{bmatrix},$$

$$\Lambda = \begin{bmatrix} \Lambda^{(1)} \\ \Lambda^{(2)} \\ \vdots \\ \Lambda^{(s)} \end{bmatrix}, \qquad \mathcal{M} = \begin{bmatrix} \mathcal{M}^{(1)} \\ \mathcal{M}^{(2)} \\ \vdots \\ \mathcal{M}^{(s)} \end{bmatrix}.$$

Now consider constraints in (4). From their linear structure of Equations in (4), it follows that there exists suitable matrices A and B such that they can be rewritten as

$$AX + BZ = 0,$$

where the mtrix A is such that A^TA is invertible.

Hence problem in (4) can be equivalently formulated as

$$\begin{aligned} \min_X \quad & F(X) \\ \text{subject to} \quad & AX + BZ = 0 \end{aligned} \tag{5}$$

where $F(X) = \sum_{i=1}^s J_i(X^{(i)})$ is a convex function in X. Observe that, from Assumption 1 and from the connectness of the graph $\mathcal{G}$, it follows that Problem in 5 admits an unique solution $\bar{X}$ such that $\bar{x}_i^{(i)} = \bar{x}_i^{(i)}$, for all $j \in \mathcal{N}_i$, $i \in V$.

Problem in (5) can be solved by the standard ADMM algorithm illustrated in [1] which consists on the following three steps

(i) Dual ascent step on the Λ and $\mathcal{M}$ variables:

$$\begin{bmatrix} \Lambda(t+1) \\ \mathcal{M}(t+1) \end{bmatrix} = \begin{bmatrix} \Lambda(t) \\ \mathcal{M}(t) \end{bmatrix} + \rho\left(AX(t) + BZ(t)\right)$$

(ii) Update of X variable:

$$X(t+1) = \operatorname*{argmin}_X \{F(X) + \left[\Lambda^T(t+1)\ \ \mathcal{M}^T(t+1)\right](AX + BZ(t))\}$$

(iii) Update of Z variable:

$$Z(t+1) = \operatorname*{argmin}_Z \left\{\left[\Lambda^T(t+1)\ \ \mathcal{M}^T(t+1)\right](AX(t+1) + BZ)\right\}$$

It is easy to see that the above steps correspond to the steps (i), (ii), (iii) of the partition-based ADMM algorithm previously described.

Proposition 4.2 in [1] guarantees, that under the assumptions that F is convex and the matrix A^TA is invertible, the trajectory $t \to X(t)$ converges to the optimal solution $\bar{X}$. This concludes the proof. ■

Observe that, in order to perform step (i) and step (ii), node i has to receive from its neighbors the information $\left\{Z^{(j)}(t)\right\}_{j\in\mathcal{N}_i}$, while, in order to perform step (iii), it has to receive the information $\left\{X^{(j)}(t+1), \Lambda^{(j)}(t+1), \mathcal{M}^{(j)}(t+1)\right\}_{j\in\mathcal{N}_i}$. Specifically, during each iteration of the partition-based ADMM scheme above described, two communication rounds between neighboring nodes have to take place in order to complete the updating actions, one before updating the multipliers $\Lambda' s$, $\mathcal{M}' s$ and the $X' s$ variables and the other before updating the $Z' s$ variables.

III. A PARTITION-BASED ADMM ALGORITHM FOR QUADRATIC FUNCTIONS

However, for the case where the functions $J_i' s$ have the particular quadratic structure illustrated in (3), the above iterations can be greatly simplified. Indeed in this case the partition-based ADMM algorithm reduces to a linear algorithm requiring, during each iteration of its implementation, only one communication round involving the $X' s$ variables. To show that, we need to introduce some auxiliary variables. Consider node i and, without loss of generality, assume $\mathcal{N}_i = \left\{j_1, \ldots, j_{|\mathcal{N}_i|}\right\}$. Then let

$$A_i = \left[A_{ii}\ A_{ij_1}\ \ldots\ A_{ij_{|\mathcal{N}_i|}}\right],$$

$$M_i = \operatorname{diag}\left\{|\mathcal{N}_i|\, I_{m_i}, I_{m_{j_1}}, \ldots, I_{m_{j_{|\mathcal{N}_i|}}}\right\}$$

$$G^{(i)} = \begin{bmatrix} G_i^{(i)} \\ G_{j_1}^{(i)} \\ \vdots \\ G_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix},\ F^{(i)} = \begin{bmatrix} F_i^{(i)} \\ F_{j_1}^{(i)} \\ \vdots \\ F_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix},\ B^{(i)} = \begin{bmatrix} B_i^{(i)} \\ B_{j_1}^{(i)} \\ \vdots \\ B_{j_{|\mathcal{N}_i|}}^{(i)} \end{bmatrix}$$

where $G_i^{(i)}, F_i^{(i)}, B_i^{(i)} \in \mathbb{R}^{m_i}$ and $G_{j_h}^{(i)}, F_{j_h}^{(i)}, B_{j_h}^{(i)} \in \mathbb{R}^{m_{j_h}}$. It turns out that $A_i \in \mathbb{R}^{r_i \times \gamma_i}$, $M_i \in \mathbb{R}^{\gamma_i \times \gamma_i}$ and $G^{(i)}, F^{(i)}, B^{(i)} \in \mathbb{R}^{\gamma_i}$, where $\gamma_i = m_i + \sum_{h=1}^{|\mathcal{N}_i|} m_{j_h}$.

The *partition-based ADMM algorithm* for quadratic functions is formally described as follows. The standing assumption is that all the matrices $A_i^T Q_i A_i + M_i$, $i \in \{1, \ldots, n\}$ are invertible.

Processor states: For $i \in \{1, \ldots, s\}$, node i stores a copy of the variables $X^{(i)}, G^{(i)}, F^{(i)}, B^{(i)}$.

Initialization: Every node initializes the variables it stores in memory to 0.

Transmission iteration: For $t \in \mathbb{N}$, at the start of the t-th iteration of the algorithm, node i transmits to node j, $j \in \mathcal{N}_i$, its estimates $x_i^{(i)}(t)$, $x_j^{(i)}(t)$. It also gathers the t-th estimates of its neighbors, $x_j^{(j)}(t)$, $x_i^{(j)}(t)$, $j \in \mathcal{N}_i$.

Update iteration: For $t \in \mathbb{N}$, node i, $i \in \{1, \ldots, s\}$, based on the information received from its neighbors, perform the following actions in order:

1) it computes $G^{(i)}(t+1)$ by setting

$$G_i^{(i)}(t) = \frac{\rho}{2} \sum_{j\in\mathcal{N}_i} \left(x_i^{(i)}(t) - x_i^{(j)}(t)\right)$$

$$G_{j_h}^{(i)}(t) = \frac{\rho}{2}\left(x_{j_h}^{(i)} - x_{j_h}^{(j_h)}\right), \quad 1 \le h \le |\mathcal{N}_i|$$

2) it computes $F^{(i)}(t+1)$ by

$$F^{(i)}(t+1) = F^{(i)}(t) + G^{(i)}(t)$$

3) it computes $B^{(i)}(t+1)$ by

$$B^{(i)}(t+1) = 2\rho M_i X^{(i)}(t) - G^{(i)}(t+1) - 2F^{(i)}(t+1)$$

4) it updates $X^{(i)}$ as follows

$$X^{(i)}(t+1) = \left[A_i^T Q_i A_i + M_i\right]^{-1}\left[A_i^T Q_i z_i + \frac{1}{2} B^{(i)}(t+1)\right]$$

The following proposition characterizes the performance of the above algorithm.

Proposition 2. *Consider the partition-based ADMM algorithm described above. Let ρ be any real number. Assume that the matrices $A_i^T Q_i A_i + M_i$, $i \in \{1, \ldots, s\}$, are invertible. Then the trajectory $t \to \left\{X^{(i)}(t)\right\}$ converge exponentially to the optimal solution, namely, for $i \in \{1, \ldots, n\}$, $x_j^{(i)}(t) \to x_j^*$ for all $j \in \mathcal{N}_i$ and, in particular,*

$$x_i^{(i)}(t) \to x_i^*.$$

The proof is based on proving that the simplified ADMM partition-based algorithm illustrated above is equivalent to the partition-based ADMM algorithm described in Section II. To do so, we next introduce the following lemmas.

Lemma 1. *The update of the variable $z_k^{(i,j)}$, $k \in \{i, j\}$, is given by*

$$z_k^{(i,j)}(t+1) = \frac{\left(\lambda_k^{(i,j)}(t+1)\right)^T + \left(\mu_k^{(j,i)}(t+1)\right)^T}{2\rho} + \frac{x_k^{(i)}(t+1) + x_k^{(j)}(t+1)}{2}$$

Proof: Without loss of generality assume that $k = i$. The value $z_i^{(i,j)}(t+1)$ is computed by setting to zero the gradient of the function

$$
\begin{aligned}
&f(z_i^{(i,j)}) \\
&\quad = \lambda_i^{(i,j)}(t+1)\left(x_i^{(i)}(t+1) - z_i^{(i,j)}\right) + \\
&\qquad + \mu_i^{(j,i)}(t+1)\left(x_i^{(j)}(t+1) - z_i^{(i,j)}\right) + \\
&\qquad + \frac{\rho}{2}\|x_i^{(i)}(t+1) - z_i^{(i,j)}\|^2 + \frac{\rho}{2}\|x_i^{(j)}(t+1) - z_i^{(i,j)}\|^2.
\end{aligned}
$$

We have

$$
\begin{aligned}
\frac{\partial f(z_i^{(i,j)})}{\partial z_i^{(i,j)}} &= -\lambda_i^{(i,j)}(t+1) - \mu_i^{(j,i)}(t+1) \\
&\quad - \rho\left(x_i^{(i)}(t+1) - z_i^{(i,j)}\right) - \rho\left(x_i^{(j)}(t+1) - z_i^{(i,j)}\right)
\end{aligned}
$$

From $\frac{\partial f(z_i^{(i,j)})}{\partial z_i^{(i,j)}} = 0$ we get the statement of the Lemma. ∎

Lemma 2. *If $\lambda_k^{(i,j)}(0) = -\mu_k^{(j,i)}(0)$, $k \in \{i,j\}$, then*

$$
\lambda_k^{(i,j)}(t) = -\mu_k^{(j,i)}(t),
$$

for $t > 0$.

Proof: The statement of the Lemma can be proved by induction. Let $\lambda_k^{(i,j)}(\ell) = -\mu_k^{(j,i)}(\ell)$, for $\ell = 0, \dots, t-1$. Then the updates take the form

$$
\begin{aligned}
&\lambda_k^{(i,j)}(t) \\
&= \lambda_k^{(i,j)}(t-1) + \rho\left(x_k^{(i)}(t-1) - z_k^{(i,j)}(t-1)\right)^T \\
&= \lambda_k^{(i,j)}(t-1) + \\
&\quad \rho\left(\left(x_k^{(i)}(t-1)\right)^T - \frac{\lambda_k^{(i,j)}(t-1) + \mu_k^{(j,i)}(t-1)}{2\rho}\right. \\
&\qquad \left. - \frac{\left(x_k^i(t-1) + x_k^{(j)}(t-1)\right)^T}{2}\right) \\
&= \lambda_k^{(i,j)}(t-1) + \rho\frac{\left(x_k^i(t-1) - x_k^{(j)}(t-1)\right)^T}{2}
\end{aligned}
$$

where the second equality follows from the previous Lemma, while the second equality comes from the inductive hypothesis. In a similar way one can obtain

$$
\mu_k^{(j,i)}(t) = \mu_k^{(j,i)}(t-1) + \rho\frac{\left(x_k^j(t-1) - x_k^{(i)}(t-1)\right)^T}{2},
$$

that, together with the inductive hypothesis, implies that $\lambda_k^{(i,j)}(t) = -\mu_k^{(j,i)}(t)$. ∎

Lemma 3. *If $\lambda_k^{(i,j)}(0) = -\mu_k^{(j,i)}(0)$, $k \in \{i,j\}$, then*

$$
z_k^{(i,j)}(t) = z_k^{(j,i)}(t),
$$

for $t \geq 0$.

Proof: From Lemma 1 and Lemma 2, we have

$$
\begin{aligned}
z_k^{(i,j)}(t) &= \frac{\left(\lambda_k^{(i,j)}(t)\right)^T + \left(\mu_k^{(j,i)}(t)\right)^T}{2\rho} \\
&\qquad + \frac{x_k^{(i)}(t) + x_k^{(j)}(t)}{2} \\
&= \frac{x_k^{(i)}(t) + x_k^{(j)}(t)}{2} = z_k^{(j,i)}(t)
\end{aligned}
$$

∎

Lemma 4. *If $\lambda_k^{(i,j)}(0) = \mu_k^{(i,j)}(0)$, $k \in \{i,j\}$, then*

$$
\lambda_k^{(i,j)}(t) = \mu_k^{(i,j)}(t),
$$

for $t \geq 0$.

Proof: The Lemma can be prove by induction. Let us assume that $\lambda_k^{(i,j)}(\ell) = \mu_k^{(i,j)}(\ell)$ for $\ell = 0, \dots, t-1$. From Lemma 1 and Lemma 2, we have that

$$
z_k^{(i,j)}(t) = \frac{x_k^{(i)}(t) + x_k^{(j)}(t)}{2}
$$

and, in turn, that

$$
\begin{aligned}
\lambda_k^{(i,j)}(t) =& \lambda_k^{(i,j)}(t-1) + \\
&+ \rho\left(x_k^{(i)}(t-1) - \frac{x_k^{(i)}(t-1) + x_k^{(j)}(t-1)}{2}\right)^T \\
\mu_k^{(i,j)}(t) =& \mu_k^{(i,j)}(t-1) + \\
&+ \rho\left(x_k^{(i)}(t-1) - \frac{x_k^{(j)}(t-1) + x_k^{(i)}(t-1)}{2}\right)^T
\end{aligned}
$$

∎

From Lemmas 1 and 2 we get the following corollary.

Corollary 1. *If for $t \geq 0$, $\lambda_k^{(i,j)}(t) = -\mu_k^{(j,i)}(t) = \mu_k^{(i,j)}(t) = -\lambda_k^{(j,i)}(t)$, $k \in \{i,j\}$, then*

$$
\begin{aligned}
&z_k^{(i,j)}(t+1) = z_k^{(j,i)}(t+1) = \frac{x_k^{(i)}(t+1) + x_k^{(j)}(t+1)}{2}; \\
&\lambda_k^{(i,j)}(t+1) = \lambda_k^{(i,j)}(t) + \frac{\rho}{2}\left(x_k^{(i)} - x_k^{(j)}\right).
\end{aligned}
$$

The above Lemmas allow us to simplify the expression of the augmented Lagragian and, precisely, we can write that

$$
\begin{aligned}
\mathcal{L} = \sum_{i=1}^{s} &\left\{ J_i(x_i^{(i)}, \{x_j^{(i)}\}_{j\in\mathcal{N}_i}) + \right. \\
&+ \sum_{j\in\mathcal{N}_i}\left[2\lambda_i^{(i,j)}\left(x_i^{(i)} - z_i^{(i,j)}\right) + 2\lambda_j^{(i,j)}\left(x_j^{(i)} - z_j^{(i,j)}\right)\right] \\
&\left. + \rho\sum_{j\in\mathcal{N}_i}\left[\|x_i^{(i)} - z_i^{(i,j)}\|^2 + \|x_j^{(i)} - z_j^{(i,j)}\|^2\right]\right\}
\end{aligned}
$$

We have the following Lemma.

Lemma 5. *The minimization over the vector $X^{(i)}$ is given by*

$$
\begin{aligned}
X_i^{(i)}(t+1) = \underset{X^{(i)}}{\operatorname{argmin}} &\left\{ J_i(X^{(i)}) + \rho\left(X^{(i)}\right)^T M_i X^{(i)} + \right. \\
&\left. - \left(X^{(i)}\right)^T B^{(i)}(t+1) \right\}
\end{aligned}
$$

where $B^{(i)}(t+1)$ and M_i are defined as in the description of the algorithm.

Proof:

$$\begin{aligned}&\underset{X^{(i)}}{\operatorname{argmin}}\Big\{J_i(X_i^{(i)})+\\&+\sum_{j\in\mathcal{N}_i}\left[2\lambda_i^{(i,j)}\left(x_i^{(i)}-z_i^{(i,j)}\right)+2\lambda_j^{(i,j)}\left(x_j^{(i)}-z_j^{(i,j)}\right)\right]\\&+\rho\sum_{j\in\mathcal{N}_i}\left[\|x_i^{(i)}-z_i^{(i,j)}\|^2+\|x_j^{(i)}-z_j^{(i,j)}\|^2\right]\Big\}=\\&\underset{X^{(i)}}{\operatorname{argmin}}\Big\{J_i(X^{(i)})+2\left(F^{(i)}(t+1)\right)^T X^{(i)}+\\&+\rho\,|\mathcal{N}_i|\;\|x_i^{(i)}\|^2+\rho\sum_{j\in\mathcal{N}_i}\|x_j^{(i)}\|^2+\\&-2\rho\left(x_i^{(i)}\right)\sum_{j\in\mathcal{N}_i}z_i^{(i,j)}(t)-2\rho\sum_{j\in\mathcal{N}_i}\left(x_j^{(i)}\right)^T z_j^{(i,j)}(t)\Big\}\end{aligned}$$

where

$$F^{(i)}(t)=\begin{bmatrix}\left(\sum_{j\in\mathcal{N}_i}\lambda_i^{(i,j)}(t)\right)^T\\\left(\lambda_{j_1}^{(j_1,i)}(t)\right)^T\\\vdots\\\left(\lambda_{j_{|\mathcal{N}_i|}}^{(j_{|\mathcal{N}_i|},i)}(t)\right)^T\end{bmatrix}$$

Let

$$M_i=\operatorname{diag}\left\{|\mathcal{N}_i|\,I_{m_i},I_{m_{j_1}},\ldots,I_{m_{j_{|\mathcal{N}_i|}}}\right\}.$$

We have that

$$\begin{aligned}&J_i(X^{(i)})+2\left(F^{(i)}(t+1)\right)^T X^{(i)}+\\&+\rho\,|\mathcal{N}_i|\;\|x_i^{(i)}\|^2+\rho\sum_{j\in\mathcal{N}_i}\|x_j^{(i)}\|^2+\\&-2\rho\left(x_i^{(i)}\right)\sum_{j\in\mathcal{N}_i}z_i^{(i,j)}(t)-2\rho\sum_{j\in\mathcal{N}_i}\left(x_j^{(i)}\right)^T z_j^{(i,j)}(t)=\\&J_i(X^{(i)})+2\left(F^{(i)}(t+1)\right)^T X^{(i)}+\rho\left(X^{(i)}\right)^T M_iX^{(i)}+\\&-2\rho\left(x_i^{(i)}\right)^T\sum_{j\in\mathcal{N}_i}\frac{x_i^{(i)}(t)+x_i^{(j)}(t)}{2}+\\&-2\rho\sum_{j\in\mathcal{N}_i}\left(x_j^{(i)}\right)^T\frac{x_j^{(i)}(t)+x_j^{(j)}(t)}{2}\end{aligned}$$

We can write

$$\begin{aligned}&-2\rho\left(x_i^{(i)}\right)\sum_{j\in\mathcal{N}_i}\frac{x_i^{(i)}(t)+x_i^{(j)}(t)}{2}+\\&-2\rho\sum_{j\in\mathcal{N}_i}\left(x_j^{(i)}\right)^T\frac{x_i^{(i)}(t)+x_i^{(j)}(t)}{2}=\\&-\rho\left(X^{(i)}\right)^T M_iX^{(i)}(t)-\rho\left(x_i^{(i)}\right)^T\sum_{j\in\mathcal{N}_i}x_i^{(j)}(t)+\\&-\rho\sum_{j\in\mathcal{N}_i}\left(x_j^{(i)}\right)^T x_j^{(j)}(t)=\\&-2\rho\left(X^{(i)}\right)^T M_iX^{(i)}(t)+\\&-\rho\left(x_i^{(i)}\right)^T\sum_{j\in\mathcal{N}_i}\left(x_i^{(j)}(t)-x_i^{(i)}(t)\right)+\\&-\rho\sum_{j\in\mathcal{N}_i}\left(x_j^{(i)}\right)^T\left(x_j^{(j)}(t)-x_j^{(i)}(t)\right)=\\&-2\rho\left(X^{(i)}\right)^T M_iX^{(i)}(t)+\left(X_i^{(i)}\right)^T G^{(i)}(t)\end{aligned}$$

where $G^{(i)}$ is defined as

$$\begin{aligned}&G_i^{(i)}(t)=\rho\sum_{j\in\mathcal{N}_i}\left(x_i^{(i)}(t)-x_i^{(j)}(t)\right)\\&G_{j_h}^{(i)}(t)=\rho\left(x_{j_h}^{(i)}(t)-x_{j_h}^{(j_h)(t)}\right),\quad 1\le h\le|\mathcal{N}_i|\end{aligned}$$

Summarizing we have that

$$\begin{aligned}X_i^{(i)}(t+1)=&\underset{X^{(i)}}{\operatorname{argmin}}\Big\{J_i(X^{(i)})+\rho\left(X^{(i)}\right)^T M_iX^{(i)}+\\&+\left(2F^{(i)}(t+1)\right)^T X^{(i)}-2\rho\left(X^{(i)}\right)^T M_iX^{(i)}(t)+\\&+\left(X_i^{(i)}\right)^T G^{(i)}(t)\Big\}.\end{aligned}$$

Hence

$$\begin{aligned}X_i^{(i)}(t+1)=&\underset{X^{(i)}}{\operatorname{argmin}}\Big\{J_i(X^{(i)})+\rho\left(X^{(i)}\right)^T M_iX^{(i)}+\\&-\left(X^{(i)}\right)^T B^{(i)}(t+1)\Big\}\end{aligned}$$

where

$$B^{(i)}(t+1)=2\rho M_iX^{(i)}(t)-G^{(i)}(t)-2F^{(i)}(t+1)$$

■

References

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.