Real-Time Distributed Control Systems 2017/18
Chapter 18
Solution of Distributed Optimization Problems: The consensus algorithm

Alexandre Bernardino

November 14, 2018

1 The global optimization problem

Our overal cost to optimize is:

$$\mathbf{d}^* = \underset{\mathbf{d}\in\mathcal{C}}{\operatorname{argmin}} \left\{\mathbf{c}^T\mathbf{d}\right\}$$

with:

$$\mathbf{d} = \left[\begin{array}{ccc} d_1 & \cdots & d_N \end{array}\right]^T$$
$$\mathbf{c} = \left[\begin{array}{ccc} c_1 & \cdots & c_N \end{array}\right]^T$$

and the constraints:

$$\mathcal{C} : \left\{ 0 \le d_i \le 100 \quad \text{and} \quad \sum_{j=1}^{N} k_{ij} d_j \ge L_i - o_i, \quad \forall i \right\}$$

2 Decomposition

Our problem can be written as:

$$\mathbf{d}^* = \underset{\mathbf{d}\in\mathcal{C}}{\operatorname{argmin}} \left\{ \sum_{i}^{N} \mathbf{c}_i^T \mathbf{d} \right\}$$

where

$$\mathbf{c}_i = \left[\begin{array}{ccccc} 0 & \cdots & c_i & \cdots & 0 \end{array}\right]^T$$

Let us define:

$$f_i(\mathbf{d}) = \begin{cases} \mathbf{c}_i^T\mathbf{d} & \textit{if} \quad \left\{ 0 \le d_i \le 100 \quad \text{and} \quad \sum_{j=1}^{N} k_{ij} d_j \ge L_i - o_i \right\} \\ +\inf & \textit{otherwise} \end{cases}$$

Thus, we can write:

$$\mathbf{d}^* = \operatorname{argmin}\{f_1(\mathbf{d}) + \cdots + f_N(\mathbf{d})\}$$

This is a case of "unconstrained" optimization, where the constraints are embedded in the individual cost functions.

3 The consensus algorithm

The consensus algorithm distributes the computation over a set of nodes in a network (see [1] for the details). Each node i holds a local copy $\mathbf{d}_i$ of the global variable $\mathbf{d}$.

$$\mathbf{d}_i = \begin{bmatrix} d_{i1} & \cdots & d_{ii} & \cdots & d_{iN} \end{bmatrix}^T$$

The cost function and the constraints of each node are partitions of the global cost function. The consensus algorithm uses a method called ADMM (alternated direction method of multipliers) with the constraints that all local copies $\mathbf{d}_i$ should be identical (thus identical to their average). ADMM uses an augmented Lagrangian method (augments the Lagrangian with a quadratic term to promote the convergence of the method) to iterate the computation of the local solutions and the Lagrange multipliers.

Beyond its local copy of the solution, $\mathbf{d}_i$, each node i maintains at each time a local copy with the average solution of all nodes, $\bar{\mathbf{d}}_i$, and a local variable of Lagrange multipliers $\mathbf{y}_i$. Then, it performs the following iterations until convergence:

1. $\mathbf{d}_i(t+1) = \underset{\mathbf{d}_i \in \mathcal{C}_i}{\operatorname{argmin}} \left\{\mathbf{c}_i^T \mathbf{d}_i + \mathbf{y}_i^T(t)(\mathbf{d}_i - \bar{\mathbf{d}}_i(t)) + \frac{\rho}{2}\|\mathbf{d}_i - \bar{\mathbf{d}}_i(t)\|_2^2\right\}$
2. $\bar{\mathbf{d}}_i(t+1) = \sum_{j=1}^N \mathbf{d}_j(t+1)$
3. $\mathbf{y}_i(t+1) = \mathbf{y}_i(t) + \rho\left(\mathbf{d}_i(t+1) - \bar{\mathbf{d}}_i(t+1)\right)$

where t is the iteration index and ρ is the augmented Lagrangian method penalty parameter.

The local constraint also only considers a reduced number of constraints (in our case the ones that constrain the illuminance level at the node's desk), so that each local problem is easy to solve:

$$\mathcal{C}_i : \left\{0 \leq d_{ii} \leq 100 \quad \text{and} \quad \mathbf{k}_i^T \mathbf{d}_i \geq L_i - o_i\right\}$$

with

$$\mathbf{k}_i = \begin{bmatrix} k_{i1} & \cdots & k_{ii} & \cdots & k_{iN} \end{bmatrix}^T$$

So, each node i needs to know:

- The local illuminance lower bound (occupancy state) L_i.
- The local external illuminance influence o_i.

- The local cost c_i.
- The local actuator bounds (in our case are all the same, 0 and 100).
- The coupling gains from the other luminaires to itself, $\mathbf{k}_i$.
- The global optimization parameter ρ

4 Feasibility check

During the computation of possible solutions, each node must check for their local feasibility. The feasibility set is an affine inequality:

$$\mathcal{C}_i : \{C_i \mathbf{d}_i \leq \mathbf{b}_i\}$$

with

$$C_i = \begin{bmatrix} -\mathbf{k}_i^T \\ -\mathbf{e}_i^T \\ \mathbf{e}_i^T \end{bmatrix}$$

where

$$\mathbf{e}_i^T = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}$$

and

$$\mathbf{b}_i = \begin{bmatrix} o_i - L_i \\ 0 \\ 100 \end{bmatrix}$$

So the feasibility check corresponds to verify if the possible solution $\mathbf{d}_i$ verifies the inequality:

$$C_i \mathbf{d}_i \leq \mathbf{b}_i \tag{1}$$

5 The Primal Iterates

To solve the problem we have to iterate on:

- the computation of the primal variables $\mathbf{d}_i$,
- the computation of the average of the primal variables of all nodes $\bar{\mathbf{d}}_i$, and
- the dual variables $\mathbf{y}_i$.

The hardest problem is the first one. Let us see how to solve it.

The primal iteration is:

$$\mathbf{d}_i(t+1) = \underset{\mathbf{d}_i \in \mathcal{C}_i}{\operatorname{argmin}} \left\{ \mathbf{c}_i^T \mathbf{d}_i + \mathbf{y}_i^T(t)(\mathbf{d}_i - \bar{\mathbf{d}}_i(t)) + \frac{\rho}{2} \|\mathbf{d}_i - \bar{\mathbf{d}}_i(t)\|_2^2 \right\}$$

It can be written as:

$$\mathbf{d}_i(t+1) = \underset{\mathbf{d}_i \in \mathcal{C}_i}{\operatorname{argmin}} \left\{ \mathbf{d}_i^T \mathbf{c}_i + \mathbf{d}_i^T \mathbf{y}_i(t) - \bar{\mathbf{d}}_i^T(t) \mathbf{y}_i(t) + \frac{\rho}{2} (\mathbf{d}_i - \bar{\mathbf{d}}(t))^T (\mathbf{d}_i - \bar{\mathbf{d}}(t)) \right\}$$

Because we are minimizing in $\mathbf{d}_i$ we can remove from the cost all terms that do not depend in $\mathbf{d}_i$.

$$\mathbf{d}_i(t+1) = \underset{\mathbf{d}_i \in \mathcal{C}_i}{\operatorname{argmin}} \left\{ \mathbf{d}_i^T \mathbf{c}_i + \mathbf{d}_i^T \mathbf{y}_i(t) + \frac{\rho}{2} \mathbf{d}_i^T \mathbf{d}_i - \rho \mathbf{d}_i^T \bar{\mathbf{d}}_i(t) \right\}$$

Now, collecting terms, we have:

$$\mathbf{d}_i(t+1) = \underset{\mathbf{d}_i \in \mathcal{C}_i}{\operatorname{argmin}} \left\{ \frac{1}{2} \rho \mathbf{d}_i^T \mathbf{d}_i + \mathbf{d}_i^T \left(\mathbf{c}_i + \mathbf{y}_i(t) - \rho \bar{\mathbf{d}}_i(t) \right) \right\}$$

Let

$$\mathbf{z}_i(t) = \rho \bar{\mathbf{d}}_i(t) - \mathbf{c}_i - \mathbf{y}_i(t)$$

Finally, we get:

$$\mathbf{d}_i(t+1) = \underset{\mathbf{d}_i \in \mathcal{C}_i}{\operatorname{argmin}} \left\{ \frac{1}{2} \rho \mathbf{d}_i^T \mathbf{d}_i - \mathbf{d}_i^T \mathbf{z}_i(t) \right\}$$

where the cost function is:

$$f^t(\mathbf{d}_i) = \frac{1}{2} \rho \mathbf{d}_i^T \mathbf{d}_i - \mathbf{d}_i^T \mathbf{z}_i(t)$$

Thus, we have an optimization problem with a quadratic function $f^t(\mathbf{d}_i)$ and convex set $\mathcal{C}_i$. The solution is either in the interior of the feasible region or at its boundary.

6 Solution in the interior

Because our cost function is a convex quadratic form, if the solution is in the interior of the set, then it is the global solution. To compute the global solution we compute for $\nabla f^t \left(\mathbf{d}_i^{(0)} \right) = 0$,

$$\nabla f^t \left(\mathbf{d}_i^{(0)} \right) = \rho \mathbf{d}_i^{(0)} - \mathbf{z}_i(t) = 0$$

The superscript (j) indicates the type of solution, (0) in the interior and other values for the boundary. Solving for $\mathbf{d}_i^{(0)}$ we get:

$$\mathbf{d}_i^{(0)}(t+1) = \rho^{-1} \mathbf{z}_i(t)$$

In terms of the individual components of the solution, we can write:

$$d_{ij}^{(0)}(t+1) = \begin{cases} \bar{d}_{ij}(t) - \rho^{-1} y_{ij}(t) & \textit{if} \quad i \neq j \\ \bar{d}_{ii}(t) - \rho^{-1} y_{ii}(t) - \rho^{-1} c_i & \textit{if} \quad i = j \end{cases}$$

In summary, the variables of the solution that are not controlled directly by the node are computed as:

$$d_{ij}^{(0)}(t+1) = \bar{d}_{ij}(t) - \rho^{-1} y_{ij}(t), \quad j \neq i$$

and the variable that is controlled by the node itself is given by:

$$d_{ii}^{(0)}(t+1) = \bar{d}_{ii}(t) - \rho^{-1}y_{ii}(t) - \rho^{-1}c_i$$

Now we must check for feasibility using Eq. (1). If the unconstrained minimum is not feasible, we have to look for the solutions on the boundary.

7 Solutions in the boundary

To look for the solutions on the boundary, we must solve the following optimization problem:

$$\mathbf{d}_i(t+1) = \underset{\mathbf{d}_i \in \partial \mathcal{C}_i}{\operatorname{argmin}} \left\{ \frac{1}{2}\rho \mathbf{d}_i^T \mathbf{d}_i - \mathbf{d}_i^T \mathbf{z}_i(t) \right\}$$

How to characterize the boundary $\partial \mathcal{C}_i$ of the feasible region $\mathcal{C}_i$?

$$\mathcal{C}_i : \{C_i \mathbf{d}_i \leq \mathbf{b}_i\}$$

The feasible region at each node i is bounded by three constraints:

- ILB – The illuminance lower bound : $\mathbf{k}_i^T \mathbf{d}_i = L_i - o_i$
- DLB –The lower bound on its dimming level : $d_{ii} = 0$
- DUB –The upper bound on its dimming level : $d_{ii} = 100$

So, the boundary of the feasible region is composed of the feasible parts of (i) the boundaries of the individual constraints ILB, DLB and DUB (hyperplanes of dimension $N-1$ - lines in a 2D problem) and (ii) the boundaries of the intersection of constraints ILB with DLB, and ILB with DUB (hyperplanes of dimension $N-2$ - points in a 2D problem). Note that DLB and DUB are impossible to hold at the same time. Mathematically the boundary of $\mathcal{C}_i$ can then be defined by the union of these five sets intersected with the feasible region:

$$\partial \mathcal{C}_i = \bigcup_{j=1}^{5} \mathcal{S}_i^{(j)} \cap \mathcal{C}_i$$

where

$$\mathcal{S}_i^{(j)} : \left\{ A_i^{(j)} \mathbf{d}_i^{(j)} = \mathbf{u}_i^j \right\}$$

with

- ILB: $A_i^{(1)} = -\mathbf{k}_i^T, \quad \mathbf{u}_i^{(1)} = o_i - L_i$
- DLB: $A_i^{(2)} = -\mathbf{e}_i^T, \quad \mathbf{u}_i^{(2)} = 0$
- DUB: $A_i^{(3)} = \mathbf{e}_i^T, \quad \mathbf{u}_i^{(3)} = 100$

- ILB $\cap$ DLB: $A_i^{(4)} = [-\mathbf{k}_i \mid -\mathbf{e}_i]^T, \quad \mathbf{u}_i^{(4)} = [o_i - L_i \mid 0]^T$
- ILB $\cap$ DUB: $A_i^{(5)} = [-\mathbf{k}_i \mid \mathbf{e}_i]^T, \quad \mathbf{u}_i^{(5)} = [o_i - L_i \mid 100]^T$

The strategy to find the solution is, thus:

1. to compute the optimum $\mathbf{d}_i^{(j)}$ of the cost function subject to each of the sets $\mathcal{S}_i^{(j)}$, then
2. check if the solution is feasible using (1), and finally
3. select the minimum among the several feasible solutions computed.

7.1 Quadratic Program with Equality Constraints

Each of the individual problems to find solutions in the boundary of the feasible set is a Quadratic Program with Equality Constraints of the form:

$$\begin{aligned} &\mathbf{d}_i(t+1) = \underset{\mathbf{d}_i}{\operatorname{argmin}} \left\{ \tfrac{1}{2}\rho \mathbf{d}_i^T \mathbf{d}_i - \mathbf{d}_i^T \mathbf{z}_i(t) \right\} \\ &s.t. \quad A_i \mathbf{d}_i = \mathbf{u}_i \end{aligned}$$

The solution can be obtained in closed form as follows. First, let us form the Lagrangian:

$$L(\mathbf{d}_i, \lambda) = \frac{1}{2}\rho \mathbf{d}_i^T \mathbf{d}_i - \mathbf{d}_i^T \mathbf{z}_i(t) + \lambda^{\mathbf{T}} (A_i \mathbf{d}_i - \mathbf{u}_i)$$

The solution is obtained at a stationary point of the Lagrangian restricted to the constraint:

$$\begin{cases} \nabla L = 0 \\ A_i \mathbf{d}_i = \mathbf{u}_i \end{cases}$$

$$\begin{cases} \rho \mathbf{d}_i - \mathbf{z}_i(t) + A_i^T \lambda = 0 \\ A_i \mathbf{d}_i = \mathbf{u}_i \end{cases}$$

$$\begin{bmatrix} \rho \mathbf{I} & A_i^T \\ A_i & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{d}_i \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{z}_i(t) \\ \mathbf{u}_i \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{d}_i \\ \lambda \end{bmatrix} = \begin{bmatrix} \rho \mathbf{I} & A_i^T \\ A_i & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z}_i(t) \\ \mathbf{u}_i \end{bmatrix}$$

To compute the matrix inverse in the previous equation let us use the following lemma:

$$\begin{bmatrix} A & B \\ B^T & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - B^T A^{-1} B)^{-1} B^T A^{-1} & A^{-1}B(D - B^T A^{-1} B)^{-1} \\ (D - B^T A^{-1} B)^{-1} B^T A^{-1} & (D - B^T A^{-1} B)^{-1} \end{bmatrix}$$

Replacing A by $\rho \mathbf{I}$, B by A_i^T, D by 0, we have:

$$\begin{bmatrix} \rho \mathbf{I} & A_i^T \\ A_i & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} \rho^{-1}\mathbf{I} - \rho^{-1} A_i^T (A_i A_i^T)^{-1} A_i & A_i^T (A_i A_i^T)^{-1} \\ (A_i A_i^T)^{-1} A_i & -\rho (A_i A_i^T)^{-1} \end{bmatrix}$$

Thus, $\mathbf{d}_i^*$ can be computed by:

$$\mathbf{d}_i^*(t+1) = \rho^{-1}\mathbf{z}_i(t) - \rho^{-1}A_i^T(A_iA_i^T)^{-1}A_i\mathbf{z}_i(t) + A_i^T(A_iA_i^T)^{-1}\mathbf{u}_i$$

Let:

$$X_i = A_iA_i^T$$

$$\mathbf{w}_i(t) = A_i\mathbf{z}_i(t)$$

Then we have the slightly simpler form:

$$\mathbf{d}_i^*(t+1) = \rho^{-1}\mathbf{z}_i(t) + A_i^T X_i^{-1}(\mathbf{u}_i - \rho^{-1}\mathbf{w}_i(t)) \tag{2}$$

7.2 Solution in $\mathcal{S}_i^1$

$$A_i^1 = -\mathbf{k}_i^T, \quad \mathbf{u}_i^1 = o_i - L_i, \quad X_i = \mathbf{k}_i^T\mathbf{k}_i = ||\mathbf{k}_i||^2, \quad \mathbf{w}_i(t) = -\mathbf{k}_i^T\mathbf{z}_i(t)$$

Replacing in (2):

$$\mathbf{d}_i^*(t+1) = \rho^{-1}\mathbf{z}_i(t) - \frac{\mathbf{k}_i}{||\mathbf{k}_i||^2}(o_i - L_i + \rho^{-1}\mathbf{k}_i^T\mathbf{z}_i(t))$$

7.3 Solution in $\mathcal{S}_i^2$

$$A_i^2 = -\mathbf{e}_i^T, \quad \mathbf{u}_i^2 = 0, \quad X_i = \mathbf{e}_i^T\mathbf{e}_i = 1, \quad \mathbf{w}_i(t) = -z_{ii}(t)$$

Replacing in (2):

$$\mathbf{d}_i^*(t+1) = \rho^{-1}\mathbf{z}_i(t) - \rho^{-1}\mathbf{e}_i z_{ii}(t)$$

In individual entries, we have:

$$d_{ij}(t+1) = \begin{cases} \rho^{-1}z_{ij}(t), & \text{if} \quad i \neq j \\ 0, & \text{if} \quad i = j \end{cases}$$

7.4 Solution in $\mathcal{S}_i^3$

$$A_i^3 = \mathbf{e}_i^T, \quad \mathbf{u}_i^3 = 100, \quad X_i = \mathbf{e}_i^T\mathbf{e}_i = 1, \quad \mathbf{w}_i(t) = z_{ii}(t)$$

Replacing in (2):

$$\mathbf{d}_i^*(t+1) = \rho^{-1}\mathbf{z}_i(t) + \mathbf{e}_i\left(100 - \rho^{-1}z_{ii}(t)\right)$$

In individual entries, we have:

$$d_{ij}(t+1) = \begin{cases} \rho^{-1}z_{ij}(t), & \text{if} \quad i \neq j \\ 100, & \text{if} \quad i = j \end{cases}$$

7.5 Solution in $\mathcal{S}_i^4$

$$A_i^4 = \begin{bmatrix} -\mathbf{k}_i^T \\ -\mathbf{e}_i^T \end{bmatrix} \quad \mathbf{u}_i^4 = \begin{bmatrix} o_i - L_i \\ 0 \end{bmatrix}, \quad X_i = \begin{bmatrix} ||\mathbf{k}_i||^2 & k_{ii} \\ k_{ii} & 1 \end{bmatrix}, \quad \mathbf{w}_i(t) = \begin{bmatrix} -\mathbf{k}_i^T \mathbf{z}_i(t) \\ -z_{ii} \end{bmatrix}$$

$$X_i^{-1} = \frac{1}{||\mathbf{k}_i||^2 - k_{ii}^2} \begin{bmatrix} 1 & -k_{ii} \\ -k_{ii} & ||\mathbf{k}_i||^2 \end{bmatrix}$$

$$A_i^T X_i^{-1} = \frac{1}{||\mathbf{k}_i||^2 - k_{ii}^2} \left[-\mathbf{k}_i + \mathbf{e}_i k_{ii} \quad | \quad \mathbf{k}_i k_{ii} - \mathbf{e}_i ||\mathbf{k}_i||^2 \right]$$

Replacing in (2):

$$\mathbf{d}_i^*(t+1) = \rho^{-1}\mathbf{z}_i(t) + \frac{-\mathbf{k}_i + \mathbf{e}_i k_{ii}}{||\mathbf{k}_i||^2 - k_{ii}^2}(o_i - L_i) - \rho^{-1}\frac{(-\mathbf{k}_i + \mathbf{e}_i k_{ii})(-\mathbf{k}_i^T \mathbf{z}_i(t)) - z_{ii}(\mathbf{k}_i k_{ii} - \mathbf{e}_i ||\mathbf{k}_i||^2)}{||\mathbf{k}_i||^2 - k_{ii}^2}$$

In individual entries, we have:

$$d_{ij}(t+1) = \begin{cases} \rho^{-1} z_{ij}(t) - \frac{k_{ij}}{||\mathbf{k}_i||^2 - k_{ii}^2}(o_i - L_i) - \rho^{-1}\frac{-k_{ij}}{||\mathbf{k}_i||^2 - k_{ii}^2}(-\mathbf{k}_i^T \mathbf{z}_i + k_{ii} z_{ii}), & \text{if} \quad i \neq j \\ 0, & \text{if} \quad i = j \end{cases}$$

7.6 Solution in $\mathcal{S}_i^5$

$$A_i^5 = \begin{bmatrix} -\mathbf{k}_i^T \\ \mathbf{e}_i^T \end{bmatrix}, \quad \mathbf{u}_i^5 = \begin{bmatrix} o_i - L_i \\ 100 \end{bmatrix}, \quad X_i = \begin{bmatrix} ||\mathbf{k}_i||^2 & -k_{ii} \\ -k_{ii} & 1 \end{bmatrix}, \quad \mathbf{w}_i(t) = \begin{bmatrix} -\mathbf{k}_i^T \mathbf{z}_i(t) \\ z_{ii} \end{bmatrix}$$

$$X_i^{-1} = \frac{1}{||\mathbf{k}_i||^2 - k_{ii}^2} \begin{bmatrix} 1 & k_{ii} \\ k_{ii} & ||\mathbf{k}_i||^2 \end{bmatrix}$$

$$A_i^T X_i^{-1} = \frac{1}{||\mathbf{k}_i||^2 - k_{ii}^2} \left[-\mathbf{k}_i + \mathbf{e}_i k_{ii} \quad | \quad -\mathbf{k}_i k_{ii} + \mathbf{e}_i ||\mathbf{k}_i||^2 \right]$$

Replacing in (2):

$$\begin{aligned} \mathbf{d}_i^*(t+1) = \rho^{-1}\mathbf{z}_i(t) &+ \frac{(-\mathbf{k}_i + \mathbf{e}_i k_{ii})(o_i - L_i) + 100(-\mathbf{k}_i k_{ii} + \mathbf{e}_i ||\mathbf{k}_i||^2)}{||\mathbf{k}_i||^2 - k_{ii}^2} \\ &- \rho^{-1}\frac{(-\mathbf{k}_i + \mathbf{e}_i k_{ii})(-\mathbf{k}_i^T \mathbf{z}_i(t)) + z_{ii}(-\mathbf{k}_i k_{ii} + \mathbf{e}_i ||\mathbf{k}_i||^2)}{||\mathbf{k}_i||^2 - k_{ii}^2} \end{aligned}$$

In individual entries, we have:

$$d_{ij}(t+1) = \begin{cases} \rho^{-1} z_{ij}(t) - \frac{k_{ij}(o_i - L_i) + 100 k_{ij} k_{ii}}{||\mathbf{k}_i||^2 - k_{ii}^2} - \rho^{-1}\frac{-k_{ij}}{||\mathbf{k}_i||^2 - k_{ii}^2}(-\mathbf{k}_i^T \mathbf{z}_i + k_{ii} z_{ii}), & \text{if} \quad i \neq j \\ 100, & \text{if} \quad i = j \end{cases}$$

References

[1] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning, Vol. 3, No. 1 (2010) 1122.*