Performance, Information Pattern Trade-offs and Computational Complexity Analysis of a Consensus Based Distributed Optimization Method

Alireza Farhadi
in collaboration with M. Cantoni and P. M. Dower

Department of Electrical and Electronic Engineering
The University of Melbourne

October 30, 2012
Motivation

Distributed Optimization Method

Computational Complexity Analysis

Future Work

References
Figure: An irrigation network.

**Figure:** An automated irrigation network via distributed distant downstream feedback control.

\[ z_i(s) = C_i(s)e_i(s), \quad C_i(s) = \frac{K_i T_i s + K_i}{s(T_i F_i s + T_i)}, \quad e_i = u_i - y_i. \]
Motivation

**Figure:** Downstream errors.

**Figure:** Upstream errors.
Motivation

Figure: Upstream input flows.

Figure: Upstream errors.
**Motivation**

**Figure:** An automated irrigation network via distributed distant downstream feedback and feedforward control. $z_i(s) = C_i(s)e_i(s) + f_i v_{i+1}$, $C_i(s) = \frac{K_i T_i s + K_i}{s(T_i F_i s + T_i)}$, $e_i = u_i - y_i$. 
Figure: An automated irrigation network equipped with a supervisory controller.
Figure: Computational complexity of the centralized optimization method versus the number of subsystems.
Distributed supervisory control

Figure: An automated irrigation network equipped with distributed supervisory controller.
Distributed optimization method (problem formulation)

\[ \min_{u=(u_1, \ldots, u_n)} \{ J(u_1, \ldots, u_n), \ u_i \subset \mathcal{U}_i \} \]

\[ \mathcal{U}_i \subset \mathbb{R}^{m_i}, \ \text{argmin}_{u_i} J(u_1, \ldots, u_n) \in \mathbb{R}^{Nm_i}. \]

**Figure:** Two-level architecture for exchanging information between distributed decision makers.
Distributed optimization method (steps$^1$)

$N_1 = \{S_1, S_2\}, \quad N_2 = \{S_3, S_4\}$

- **Initialization:** The information exchange between neighborhoods at outer iterate $t$ makes it possible for subsystem $S_i$ to initialize its local decision variables as $h_i^0 = u_i^t$, where $u_i^0 \in \mathcal{U}_i$ are chosen arbitrarily at time $t = 0$.

- **Inner Iterate:** Then, subsystem $S_i$ performs $\bar{p}$ inner iterates as follows:

For inner iterate $p \in \{0, 1, ..., \bar{p} - 1\}$, it first updates its decision variable via

$$h_{i}^{p+1} = \pi_i h_i^* + (1 - \pi_i) h_i^p,$$

where

$$\pi_1 + \pi_2 = 1, \quad \pi_3 + \pi_4 = 1$$

and

$$h_1^* = \arg\min_{h_1 \in \mathcal{U}_1} J(h_1, h_2^p, h_3^0, h_4^0), \quad h_2^* = \arg\min_{h_2 \in \mathcal{U}_2} J(h_1^p, h_2, h_3^0, h_4^0),$$

$$h_3^* = \arg\min_{h_3 \in \mathcal{U}_3} J(h_1^0, h_2^0, h_3, h_4^p), \quad h_4^* = \arg\min_{h_4 \in \mathcal{U}_4} J(h_1^0, h_2^0, h_3^p, h_4).$$

---

Distributed optimization method (steps)

- **Inner Iterate (continued):** Then, subsystem $S_i$ trades its updated decision variable $h_i^{p+1}$ with all other subsystems within its neighborhood.

- **Outer Iterate:** After $\bar{p}$ inner iterates there is an outer iterate update as follows

\[
u_{i}^{t+1} = \lambda_i h_i^{\bar{p}} + (1 - \lambda_i) u_i^t,
\]

where
\[
\lambda_1 = \lambda_2, \quad \lambda_3 = \lambda_4, \quad \lambda_1 + \lambda_3 = 1.
\]

Then, there is an outer iterate communication, in which the updated decision variables $u_i^{t+1}$ are shared between all neighborhoods and subsequently between all subsystems.
Feasibility, convergence and optimality results \(^2\)

**Feasibility:** Given any collection of disjoint neighborhoods, above strictly convex finite horizon cost functional \(J\), convex control constraint sets \(\mathcal{U}_i\) and a feasible initialization (i.e., \(u_i^0 \in \mathcal{U}_i\)), the inner and outer iterates are feasible (i.e., \(h_i^{p+1}, u_i^{t+1} \in \mathcal{U}_i\)).

**Convergence:** Given any collection of disjoint neighborhoods and a feasible initialization, the strictly convex finite horizon cost functional \(J(u_1^t, \ldots, u_n^t)\) is non-increasing at each outer iterate \(t\) and converges as \(t \to \infty\).

**Optimality:** Given any collection of disjoint neighborhoods, a feasible initialization, strictly convex and quadratic cost \(J\), and closed convex control constraint sets \(\mathcal{U}_i\), the cost \(J(u_1^t, \ldots, u_n^t)\) converges to the optimal cost \(J(u_1^*, \ldots, u_n^*)\), and the iterates \((u_1^t, \ldots, u_n^t)\) converge to the unique optimal solution \((u_1^*, \ldots, u_n^*)\), as \(t \to \infty\).

Interaction strength decomposition method

Figure: Left: Communication graph. Right: Interaction strength graph summarizing the effects of decision variables on subsystems.

No hopping is allowed for intra-neighborhood communication ⇒ Following the communication graph, the size of each neighborhood must be at most 2:
Option1: \{S_2, S_3\}, \{S_4, S_5\}, \{S_6, S_1\}
Option2: \{S_1, S_2\}, \{S_3, S_4\}, \{S_5, S_6\}

Following interaction strength graph, option 2 is selected.
Interaction strength decomposition method

Dynamic system:

\[ S_i : \ x_i[k + 1] = A_i x_i[k] + B_i u_i[k] + v_i[k], \ i = 1, 2, ..., n, \ k \in \{0, 1, 2, ..., N - 1\}, \]

where

\[ v_i[k] = \sum_{j=1, j \neq i}^{n} M_{ij} x_j[k] + N_{ij} u_j[k]. \]

Transfer function from \( U(z) = (U'_1(z) \ldots U'_n(z))' \) to state \( X(z) = (X'_1(z) \ldots X'_n(z))' \) is given by

\[ G(z) = V^{-1}(z) W(z), \]

where \( V(z) \coloneqq [V_{ij}(z)] \) with

\[ V_{ij}(z) \coloneqq \begin{cases} I_{n_i}, & \text{when } i = j \\ -(zI_{n_i} - A_i)^{-1} M_{ij}, & \text{otherwise} \end{cases} \]

and \( W(z) \coloneqq [W_{ij}(z)] \) with

\[ W_{ij}(z) \coloneqq \begin{cases} (zI_{n_i} - A_i)^{-1} B_i, & \text{when } i = j \\ (zI_{n_i} - A_i)^{-1} N_{ij}, & \text{otherwise}. \end{cases} \]
Interaction strength decomposition method

\[ G(z) \big|_{z=1} = \begin{pmatrix}
  E_1 & E_{12} & \cdots & E_{1n} \\
  E_{21} & E_2 & \cdots & E_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  E_{n1} & E_{n2} & \cdots & E_n
\end{pmatrix}, \quad E_{ij} \in \mathbb{R}^{n_i \times m_j}. \]

Interaction Strength (IS):

\[ IS_{ij} = \begin{cases} 
0, & \text{if } i = j \\
\frac{\sigma_{\text{max}}(E_{ij})}{\sigma_{\text{min}}(E_i)}, & \text{if } \sigma_{\text{min}}(E_i) \neq 0 \text{ and } i \neq j \\
\frac{\sigma_{\text{max}}(E_{ij})}{\gamma}, & \text{if } \sigma_{\text{min}}(E_i) = 0 \text{ and } i \neq j
\end{cases} \]

Normalized interaction strength:

\[ ISN_{ij} = \text{round}\left( \frac{IS_{ij}}{IS_{\text{min}}} \right), \quad IS_{\text{min}} = \min_{\{i,j; IS_{ij} > 0\}} IS_{ij}. \]
Interaction strength decomposition method

Example: Consider a system with six interacting scalar subsystems. The aggregated system is described as follows:

\[ x[k + 1] = Ax[k] + Bu[k], \]

\[ x[k] = (x_1[k], x_2[k], x_3[k], x_4[k], x_5[k], x_6[k])', \]

\[ u[k] = (u_1[k], u_2[k], u_3[k], u_4[k], u_5[k], u_6[k])', \]

\[
A = \begin{pmatrix}
1.7049 & -0.0049 & -0.9082 & -0.2732 & 0.5496 & -0.2756 \\
0.2328 & 1.4672 & -0.0213 & -0.4127 & -0.4861 & 0.5709 \\
0.1213 & -0.1213 & 0.7311 & 0.0955 & 0.5566 & -0.4652 \\
-0.3836 & 0.3836 & 0.1393 & 1.2061 & 0.132 & 0.198 \\
-0.1148 & 0.1148 & -0.6754 & 0.007 & 2.3762 & -0.4357 \\
-0.5148 & 0.5148 & 0.0246 & -0.143 & 0.4762 & 1.5143
\end{pmatrix},
\]

\[ B = \text{diag}(1.7, -1, 1.5, -1.2, 1.9, 0.86). \]
Interaction strength decomposition method

Interaction strength matrix:

<table>
<thead>
<tr>
<th>Subsystems</th>
<th>$S_1$</th>
<th>$S_2$</th>
<th>$S_3$</th>
<th>$S_4$</th>
<th>$S_5$</th>
<th>$S_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>0</td>
<td>36</td>
<td>226</td>
<td>3</td>
<td>245</td>
<td>82</td>
</tr>
<tr>
<td>$S_2$</td>
<td>37</td>
<td>0</td>
<td>21</td>
<td>29</td>
<td>49</td>
<td>27</td>
</tr>
<tr>
<td>$S_3$</td>
<td>20</td>
<td>12</td>
<td>0</td>
<td>22</td>
<td>182</td>
<td>70</td>
</tr>
<tr>
<td>$S_4$</td>
<td>93</td>
<td>55</td>
<td>63</td>
<td>0</td>
<td>148</td>
<td>39</td>
</tr>
<tr>
<td>$S_5$</td>
<td>53</td>
<td>31</td>
<td>151</td>
<td>13</td>
<td>0</td>
<td>67</td>
</tr>
<tr>
<td>$S_6$</td>
<td>106</td>
<td>62</td>
<td>73</td>
<td>1</td>
<td>185</td>
<td>0</td>
</tr>
</tbody>
</table>

Strength weights \((SW(ij) = ISN_{ij} + ISN_{ji}, \ i \neq j)\)

\[
\begin{align*}
(1, 2) &= 73 & (1, 3) &= 246 & (1, 4) &= 96 & (1, 5) &= 298 \\
(1, 6) &= 188 & (2, 3) &= 33 & (2, 4) &= 84 & (2, 5) &= 80 \\
(2, 6) &= 89 & (3, 4) &= 85 & (3, 5) &= 333 & (3, 6) &= 143 \\
(4, 5) &= 161 & (4, 6) &= 40 & (5, 6) &= 252 & (5, 6) &= 252 \\
\end{align*}
\]

\[N_1 = \{S_3, S_5\}, \quad N_2 = \{S_1, S_6\}, \quad N_3 = \{S_2, S_4\}.\]
Interaction strength decomposition method

Strength weights \( SW(ijk) = ISN_{ij} + ISN_{ik} + ISN_{ji} + ISN_{jk} + ISN_{ki} + ISN_{kj}, \ i \neq j \neq k \)

<table>
<thead>
<tr>
<th></th>
<th>((1, 2, 3) = 352)</th>
<th>((1, 2, 4) = 253)</th>
<th>((1, 2, 5) = 451)</th>
</tr>
</thead>
<tbody>
<tr>
<td>((1, 2, 6) = 350)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((1, 3, 6) = 577)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((1, 4, 5) = 555)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((1, 5, 6) = 738)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((2, 3, 6) = 265)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((2, 4, 5) = 325)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((2, 5, 6) = 421)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((3, 4, 5) = 579)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((3, 4, 6) = 268)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((3, 5, 6) = 728)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((4, 5, 6) = 453)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( N_1 = \{S_1, S_3, S_5\}, \quad N_2 = \{S_2, S_4, S_6\} \).
Performance criteria

**Performance Loss:** For a given number of outer iterate updates $t$ and $\bar{p}$, the Performance Loss $PL_t(\bar{p})$ (measured in percent) is defined as

$$PL_t(\bar{p}) = 100\left(\frac{J(u^t_1, \ldots, u^t_n) - \bar{J}}{\bar{J}}\right),$$

where $\bar{J}$ is the optimal cost.

**Total Number of Iterations:** For a given $\bar{p}$,

$$T_t = \bar{p} \times t$$

is referred as the total number of iterations up to outer iterate $t$.

**Total Number of Iterations for Convergence:** For a given performance loss $PL$, let $\bar{t}_{PL}$ be the smallest integer such that

$$PL_t(\bar{p}) \leq PL \quad \text{for all } t \geq \bar{t}_{PL}.$$  

Then,

$$T_{PL} = \bar{p} \times \bar{t}_{PL}$$

is referred as the total number of iterations for convergence.
Illustrative example

Dynamic system:

\[ S_i : x_i[k + 1] = A_i x_i[k] + B_i u_i[k] + v_i[k], \quad i = 1, 2, \ldots, 6, \quad k \in \{0, 1, 2, 3, 4\}, \]

where

\[ x_i[0] = 0, \quad v_i[k] = \sum_{j=1, j\neq i}^{6} M_{ij} x_j[k]. \]

\[ \min_{u} \left\{ J(x[0], u_1, \ldots, u_6), x_i[k] \in \mathcal{X}_i = [-12, 12], u_i[k] \in \mathcal{G}_i = [-6, 6], \forall i, k \right\}, \]

\[ J(x[0], u_1, \ldots, u_6) = \sum_{i=1}^{6} \sum_{k=0}^{4} ||x_i[k] - x_i^d||^2 + ||u_i[k]||^2. \]

\[ x_1^d = 1, x_2^d = 2, x_3^d = 3, x_4^d = 4, x_5^d = 5, x_6^d = 6, \]

\[ J = 9370.89. \]
### Performance, Information Pattern Trade-offs and Computational Complexity Analysis of a Consensus Based Distributed Optimization Method

#### Distributed Optimization Method

\[
\bar{p} \quad T_{PL} \quad PL_t(\bar{p}) \text{ at } t = T_{PL}/\bar{p} \quad \text{Computation time (sec.)}
\]

<table>
<thead>
<tr>
<th>(\bar{p})</th>
<th>(T_{PL})</th>
<th>(PL_t(\bar{p}) \text{ at } t = T_{PL}/\bar{p})</th>
<th>Computation time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>453</td>
<td>0.99</td>
<td>77.63</td>
</tr>
<tr>
<td>10</td>
<td>820</td>
<td>0.95</td>
<td>142.34</td>
</tr>
<tr>
<td>20</td>
<td>1400</td>
<td>0.93</td>
<td>244.93</td>
</tr>
<tr>
<td>50</td>
<td>3250</td>
<td>0.98</td>
<td>564.91</td>
</tr>
</tbody>
</table>

**Table**: Two-neighborhoods case.

<table>
<thead>
<tr>
<th>(\bar{p})</th>
<th>(T_{PL})</th>
<th>(PL_t(\bar{p}) \text{ at } t = T_{PL}/\bar{p})</th>
<th>Computation time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>424</td>
<td>0.99</td>
<td>74.23</td>
</tr>
<tr>
<td>10</td>
<td>2200</td>
<td>0.99</td>
<td>390.14</td>
</tr>
<tr>
<td>20</td>
<td>4320</td>
<td>0.98</td>
<td>755.36</td>
</tr>
<tr>
<td>50</td>
<td>10750</td>
<td>0.99</td>
<td>1885.2</td>
</tr>
</tbody>
</table>

**Table**: Three-neighborhoods case.

<table>
<thead>
<tr>
<th>(\bar{p})</th>
<th>(T_{PL})</th>
<th>(PL_t(\bar{p}) \text{ at } t = T_{PL}/\bar{p})</th>
<th>Computation time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1020</td>
<td>0.99</td>
<td>179.21</td>
</tr>
<tr>
<td>10</td>
<td>10200</td>
<td>0.99</td>
<td>1834.3</td>
</tr>
<tr>
<td>20</td>
<td>20400</td>
<td>0.99</td>
<td>3569.9</td>
</tr>
<tr>
<td>50</td>
<td>51000</td>
<td>0.99</td>
<td>9027.9</td>
</tr>
</tbody>
</table>

**Table**: Six-neighborhoods case.
Illustrative example

Figure: Computation time versus the total number of iterations for convergence $T_{PL}$ for different decompositions and $PL = 1$ percent. Red: The two-neighborhoods case. Blue: The three-neighborhoods case. Black: The six-neighborhoods case.

Computation time equals $\gamma T_{PL}$, where $\gamma = 0.175$. 
Illustrative example

Figure: Trade-offs between $PL_t(\bar{p})$ and $T_t$ for different decompositions and $\bar{p} = 10$ (top figure) and $\bar{p} = 20$ (bottom figure). Red: The two-neighborhoods case. Blue: The three-neighborhoods case. Black: The six-neighborhoods case.
Illustrative example

Figure: Trade-offs between the total number of iterations for convergence $T_{PL}$ and $\bar{p}$ for different decompositions and $PL = 1$ percent (top figure) and $PL = 10$ percent (bottom figure). Red: The two-neighborhoods case. Blue: The three-neighborhoods case. Black: The six-neighborhoods case.
Example:

Inner iterate communication overhead: 1 second

Outer iterate communication overhead: 10 seconds

For the system decomposed into 3 neighborhoods with \( \bar{p} = 10 \):

Total communication overhead equals \( (220 \times 10 + 2200 \times 1 =) 4400 \) seconds

Total computation time for producing the optimal inputs equals \( 390.14 + 4400 = ) 4790.14 \) seconds.

Without decomposition and inner iterates:

Total communication overhead equals \( 950 \times 10 = ) 9500 \) seconds

Total computation time for producing the optimal inputs equals \( 174.126 + 9500 = ) 9674.126 \) seconds.
Computational Complexity Analysis

Figure: An automated irrigation network via distributed distant downstream feedback control.

\[ z_i(s) = C_i(s)e_i(s), \quad C_i(s) = \frac{K_i T_i s + K_i}{s(T_i F_i s + T_i)}, \quad e_i = u_i - y_i. \]

Automated irrigation network model:

\[ S_i : \quad x_i[k + 1] = A_i x_i[k] + B_i u_i[k] + F_i d_i[k] + v_i[k], \quad v_i[k] = M_i x_{i+1}[k], \]
\[ y_i[k] = C_i x_i[k], \]
\[ z_i[k] = D_i x_i[k], \]
\[ i = 1, 2, \ldots, n, \quad k \in \{0, 1, 2, \ldots, N - 1\}. \]
Cost functional:

$$\min_{u=(u_1,\ldots,u_n)} \left\{ J(x[0], d, y_d, u_1, \ldots, u_n), L_i \leq y_i[k], u_i[k] \leq H_i, 0 \leq z_i[k] \leq Z_i, \forall i, k \right\},$$

$$J(x[0], d, y_d, u_1, \ldots, u_n) = \sum_{i=1}^{n} \sum_{k=0}^{N-1} ||y_i[k] - y_i^d||_Q^2 + ||z_i[k]||_P^2 + ||u_i[k] - u_i[k-1]||_R^2.$$
Centralized technique (active set method)

Number of decision variables: \( n_d \)

Number of inequality constraints: \( n_c \)

\[ C_{cen}(n_d) \sim \mathcal{O}(n_d^3), \quad \text{(for a given } n_c) \]

\[ C_{cen}(n_c) \sim \mathcal{O}(n_c^3), \quad \text{(for a given } n_d) \]

\[ C_{cen}(n_d, n_c) \sim \mathcal{O}(n_d^3 \times n_c^3) \]

For automated irrigation networks: \( n_d = nN, \; n_c = 6nN \)

\[ C_{cen}(n) \sim \mathcal{O}(n_d^3 \times n_c^3) \sim \mathcal{O}(n^6) \]

---

5 [ECC2009],[TCST2010].
Distributed technique

For synchronized communication:

\[ C_{\text{dis}}(n) = \sum_{j=1}^{T_{PL}(n)} C_j(n), \]

\[ T_{PL}(n): \text{Total number of iterations for convergence} \]
\[ C_j(n): \text{Maximum computation time of the decision maker with the dominating computational complexity} \]

**Assumption**: Distributed decision makers also use active set method for their smaller QPs.

Number of decision variables of each decision maker: \( N \)

Number of inequality constraints of the dominating decision maker:

\[
\begin{cases} 
N(4n + 1), & \text{if } n \leq \frac{N}{2} \\
N(4 \left\lfloor \frac{N}{2} \right\rfloor + 2), & \text{otherwise}
\end{cases}
\]
Distributed technique

For a given $n$, the dominating decision maker remains constant for all iterations, whereby the dominating computational complexity $C_j(n)$ also remains constant for all $j > 1$

$$C_j(n) = C(n), \quad \forall j > 1.$$ 

For $j = 1$, it takes some time that variables to be placed into the cache memory

$$C_1(n) \geq C_j(n) = C(n), \quad \forall j \geq 1.$$ 

$$C_{dis}(n) = \sum_{j=1}^{T_{PL}(n)} C_j(n) = C_1(n) + (T_{PL}(n) - 1)C(n)$$
Distributed technique

Number of inequality constraints of the dominating decision maker:

\[
\begin{cases}
    N(4n + 1), & \text{if } n \leq \frac{N}{2} \\
    N(4 \left\lfloor \frac{N}{2} \right\rfloor + 2), & \text{otherwise}
\end{cases}
\]

\[\Rightarrow\]

\[C(n) \sim \begin{cases}
    \mathcal{O}(n), & \text{if } n \leq \frac{N}{2} \\
    \alpha, & \text{otherwise}
\end{cases}\]

\[C_1(n) = \eta, \quad T_{PL}(n) = \beta n\]

\[C_{dis}(n) = C_1(n) + (T_{PL}(n) - 1)C(n) \sim \begin{cases}
    \mathcal{O}(n^2), & \text{if } n \leq \frac{N}{2} \\
    \mathcal{O}(n), & \text{otherwise}
\end{cases}\]
Simulation results

\[ C(n) \approx \begin{cases} 
0.00983n + 0.118 \sim O(n), & \text{if } n \leq 12 \\
0.269, & \text{otherwise}
\end{cases} \]

\[ T_{PL}(n) = 1.5n, \quad C_1(n) \approx C_1 = 1.36. \]
Simulation results

![Graph showing the relationship between C_{dis}(n) and n.]

\[
C_{dis}(n) = C_1(n) + (T_{PL}(n) - 1)C(n)
\]  

(1)

\[
C(n) \approx \begin{cases} 
0.00983n + 0.118 \sim O(n), & \text{if } n \leq 12 \\
0.269, & \text{otherwise}
\end{cases} \quad T_{PL}(n) = 1.5n, \quad C_1(n) \approx C_1 = 1.36.
\]

(2)

\[
C_{dis}(n) \approx \begin{cases} 
0.0147n^2 + 0.167n + 1.242 \sim O(n^2), & \text{if } n \leq 12 \\
0.403n + 1.091 \sim O(n), & \text{otherwise}
\end{cases}
\]
Simulation result

\[ C_{cen} \approx \left( \frac{n}{12} \right)^6 \sim O(n^6). \]  \hspace{1cm} (3)
Finding an analytical expression for $T_{PL}$ (and therefore $C_{dis} = \sum_{j=1}^{T_{PL}} C_j$)

$$T_{PL} = F(\lambda_{m,l}, \pi_{m,l}, PL, \bar{p}, q, l).$$

Finding an analytical expression for communication overhead: $Com$

$$Com = G(\bar{p}, q, l).$$

Balancing interactions between control, computation, communication, and scalability to have the best possible performance: good quality control inputs with minimum overall computation time

$$\min_{\lambda_{m,l}, \pi_{m,l}, PL, \bar{p}, q, l} \left\{ C_{dis} + Com, \quad \text{subject to constraints on } \lambda_{m,l}, \pi_{m,l}, PL \right\}$$

$PL$: Quality of control

$\lambda_{m,l}, \pi_{m,l}$: Convergence rate, quality of distributed computation

$\bar{p}$: Communication pattern

$q, l$: Scalability architecture

