A Universal Decomposition for Distributed Optimization Algorithms

Bryan Van Scoy  
Miami University

Laurent Lessard  
Northeastern University
Distributed optimization

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} f_i(x_i) \\
\text{subject to} & \quad x_1 = x_2 = \ldots = x_n
\end{align*}
\]

\[
x = \begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{bmatrix} \quad \nabla f = \begin{bmatrix}
\nabla f_1 \\
\nabla f_2 \\
\vdots \\
\nabla f_n
\end{bmatrix} 
\]

\[
L = \begin{bmatrix}
\frac{2}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} \\
0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 \\
0 & -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} & 0 \\
-\frac{1}{3} & 0 & -\frac{1}{3} & 1 & -\frac{1}{3} \\
-\frac{1}{3} & 0 & 0 & -\frac{1}{3} & \frac{2}{3}
\end{bmatrix}
\]
Algorithms

\[ x^{t+1} = (I - L)x^t - \alpha \nabla f(x^t) \]  
DGD’09

\[ x^{t+1} = (I - \frac{1}{2}L)(2x^t - x^{t-1}) - \alpha \nabla f(x^t) + \alpha \nabla f(x^{t-1}) \]  
EXTRA’15

\[ x^{t+1} = (I - L)(x^t - \alpha y^t) \]  
AugDGM’15

\[ y^{t+1} = (I - L)(y^t + \nabla f(x^{t+1}) - \nabla f(x^t)) \]

\[ x^{t+1} = (I - L)x^t - \alpha y^t \]  
DIGing’17

\[ y^{t+1} = (I - L)y^t + \nabla f(x^{t+1}) - \nabla f(x^t) \]  
ExDiff’17

\[ y^{t+1} = (I - \frac{1}{2}L)x^t - \alpha \nabla f((I - \frac{1}{2}L)x^t) \]  
ExDiff’17

\[ x^{t+1} = (I - \frac{1}{2}L)x^t + y^{t+1} - y^t \]  
ExDiff’17

\[ x^{t+1} = (I - \frac{1}{2}L)(2x^t - x^{t-1} - \alpha \nabla f(x^t) + \alpha \nabla f(x^{t-1})) \]  
NIDS’19

and more...
Questions

1) Does every algorithm decompose into a centralized optimization method and a consensus estimator?

2) Given a centralized optimization method and a consensus estimator, can we combine them to form an algorithm?
**Optimization**

\[ y = G_{\text{opt}} u \]
\[ u = \nabla f(y) \]

**Algorithm**

| Method               | Equation                                      | \( \hat{G}_{\text{opt}}(z) \)                  |
|----------------------|-----------------------------------------------|------------------------------------------------|
| Gradient descent     | \( y^{t+1} = y^t - \alpha \nabla f(y^t) \)   | \( \frac{-\alpha}{z - 1} \)                   |
| Accelerated methods  | \( x^{t+1} = x^t + \beta(x^t - x^{t-1}) - \alpha \nabla f(y^t) \) \[ y^t = x^t + \gamma(x^t - x^{t-1}) \] | \( \frac{-\alpha(z + \gamma(z - 1))}{(z - 1)(z - \beta)} \) |
| Proximal methods     | \( y^{t+1} \in \arg \min_y f(y) + \frac{1}{2\alpha} \|y - y^t\|^2 \) | \( \frac{-\alpha z}{z - 1} \)                 |
Consensus

\[
\begin{bmatrix}
y_i \\
z_i
\end{bmatrix} = G_{\text{con}} \begin{bmatrix} w_i \\ v_i \end{bmatrix}
\]

\[
v_i = \sum_{j=1}^{n} a_{ij} (z_i - z_j)
\]

Algorithm

\[
\begin{aligned}
\hat{G}_{\text{con}}(z) &= \\
\end{aligned}
\]

First-order estimator

\[
x^{t+1} = L(w^t - x^t) \\
y^t = w^t - x^t
\]

Second-order estimator

\[
x^{t+1} = 2x^t - x^{t-1} + Ly^t \\
y^t = w^t - \frac{1}{2}(2x^t - x^{t-1})
\]
Our approach

Define optimization methods, consensus estimators, and distributed optimization algorithms.

- correct fixed points
- convergence on quadratic functions $f(x) = \varepsilon \|x\|^2$ for small $\varepsilon$
- convergence on balanced and sufficiently connected graphs

For LTI systems, give equivalent conditions in terms of transfer functions.

- for example, $\hat{G}_{\text{opt}}$ has a pole at $z = 1$ and is proper
A universal decomposition

- $G_{\text{opt}}$ is a centralized optimization method
- $G_{\text{con}}$ is a second-order consensus estimator

Every distributed optimization algorithm decomposes in this form, and any optimization method and consensus estimator form a valid algorithm.
Non-accelerated algorithms

\[ \hat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & \frac{1-z}{(z-1)^2} \\ 1 & \frac{1-z}{(z-1)^2} \end{bmatrix} \]

\[ \hat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & -\frac{1}{2}z^2 \\ 1 & \frac{1}{2}-z \end{bmatrix} \]

\[ \hat{G}_{\text{con}}(z) = \begin{bmatrix} 1 & \frac{-z(z+\beta-1)}{(z-1)^2} \\ 1 & \frac{1-(1+\beta)z}{(z-1)^2} \end{bmatrix} \]

In all cases, \( \hat{G}_{\text{opt}}(z) = \frac{-\alpha}{z-1} \) (gradient descent).
Internal stability

\[ \hat{G}_{\text{opt}} \] must have a pole at \( z = 1 \)

- at steady-state, the gradients \( \mathbf{u} \) are non-zero, so \( \mathbf{w} \) grows linearly

The dynamics are not internally stable.
Factored form

- $G_{opt}$ is a centralized optimization method
- $G_{con1}$ and $G_{con2}$ are first-order consensus estimators

Better properties (internal stability), but not all algorithms factor.

This form was first proposed by Shuo Han (2020)
Non-accelerated algorithms that factor

\[ \hat{G}_{\text{con}1}(z), \hat{G}_{\text{con}2}(z) = \begin{bmatrix} 1 & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & \frac{-z}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix} \]

- DIGing uses the estimator on the left for both factors
- AB uses one of each estimator
- AugDGM uses the estimator on the right for both factors

In all cases, \( \hat{G}_{\text{opt}}(z) = \frac{-\alpha}{z-1} \) (gradient descent).
Accelerated algorithms that factor

\[
\hat{G}_{\text{con}1}(z) = \begin{bmatrix} 1 & \frac{1}{\alpha} \hat{G}_{\text{opt}}(z) \\ 1 & \frac{1}{\alpha} \hat{G}_{\text{opt}}(z) \end{bmatrix} \quad \text{and} \quad \hat{G}_{\text{con}2}(z) = \begin{bmatrix} 1 & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix}
\]

- \textit{ABN} uses \( \gamma = \beta \) (Nesterov’s method)
- \textit{ABm} uses \( \gamma = 0 \) (heavy-ball method)

In all cases, \( \hat{G}_{\text{opt}}(z) = \frac{-\alpha(z+\gamma(z-1))}{(z-1)(z-\beta)} \) (accelerated method).
The optimization method must be robust to dynamic uncertainty.
The consensus estimator must be stable under nonlinear feedback.
Every distributed optimization algorithm decomposes in this form, and any optimization method and consensus estimator form a valid algorithm.

- provides a systematic way to interpret and compare algorithms
- interpretations as robust optimization and consensus with feedback
- how can we leverage this decomposition to design optimal algorithms?