A Universal Decomposition for Distributed Optimization Algorithms

Bryan Van Scoy
Miami University

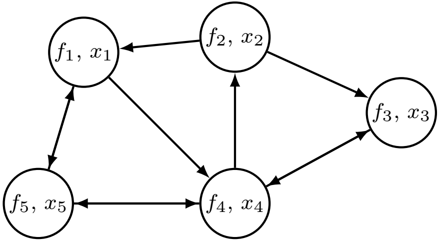
Laurent Lessard
Northeastern University

IEEE Conference on Decision and Control

December 6–9, 2022

Distributed optimization

$$\underset{x_1,\ldots,x_n}{\text{minimize}} \quad \sum_{i=1}^{n} f_i(x_i)$$
$$\text{subject to} \quad x_1 = x_2 = \ldots = x_n$$



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \nabla \mathbf{f} = \begin{bmatrix} \nabla f_1 \\ \nabla f_2 \\ \vdots \\ \nabla f_n \end{bmatrix} \qquad \mathbf{L} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & 0 & 0 & -\frac{1}{3} \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ 0 & -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} & 0 \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & -\frac{1}{3} \\ -\frac{1}{3} & 0 & 0 & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

Algorithms

$$\mathbf{x}^{t+1} = (I - \mathbf{L})\mathbf{x}^t - \alpha \nabla \mathbf{f}(\mathbf{x}^t)$$ DGD'09

$$\mathbf{x}^{t+1} = (I - \tfrac{1}{2}\mathbf{L})(2\mathbf{x}^t - \mathbf{x}^{t-1}) - \alpha \nabla \mathbf{f}(\mathbf{x}^t) + \alpha \nabla \mathbf{f}(\mathbf{x}^{t-1})$$ EXTRA'15

$$\begin{aligned} \mathbf{x}^{t+1} &= (I - \mathbf{L})(\mathbf{x}^t - \alpha \mathbf{y}^t) \\ \mathbf{y}^{t+1} &= (I - \mathbf{L})(\mathbf{y}^t + \nabla \mathbf{f}(\mathbf{x}^{t+1}) - \nabla \mathbf{f}(\mathbf{x}^t)) \end{aligned}$$ AugDGM'15

$$\begin{aligned} \mathbf{x}^{t+1} &= (I - \mathbf{L})\mathbf{x}^t - \alpha \mathbf{y}^t \\ \mathbf{y}^{t+1} &= (I - \mathbf{L})\mathbf{y}^t + \nabla \mathbf{f}(\mathbf{x}^{t+1}) - \nabla \mathbf{f}(\mathbf{x}^t) \end{aligned}$$ DIGing'17

$$\begin{aligned} \mathbf{y}^{t+1} &= (I - \tfrac{1}{2}\mathbf{L})\mathbf{x}^t - \alpha \nabla \mathbf{f}\big((I - \tfrac{1}{2}\mathbf{L})\mathbf{x}^t\big) \\ \mathbf{x}^{t+1} &= (I - \tfrac{1}{2}\mathbf{L})\mathbf{x}^t + \mathbf{y}^{t+1} - \mathbf{y}^t \end{aligned}$$ ExDiff'17

$$\mathbf{x}^{t+1} = (I - \tfrac{1}{2}\mathbf{L})\big(2\mathbf{x}^t - \mathbf{x}^{t-1} - \alpha \nabla \mathbf{f}(\mathbf{x}^t) + \alpha \nabla \mathbf{f}(\mathbf{x}^{t-1})\big)$$ NIDS'19

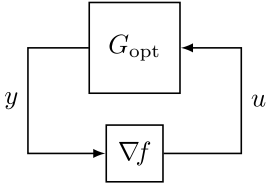
and more. . .

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?

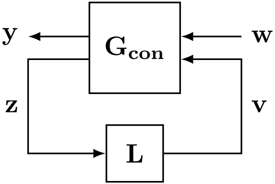
Optimization



$$y = G_{\text{opt}}\, u$$
$$u = \nabla f(y)$$

	Algorithm	$\widehat{G}_{\text{opt}}(z)$
Gradient descent	$y^{t+1} = y^t - \alpha\,\nabla f(y^t)$	$\dfrac{-\alpha}{z-1}$
Accelerated methods	$x^{t+1} = x^t + \beta(x^t - x^{t-1}) - \alpha\nabla f(y^t)$ $y^t = x^t + \gamma(x^t - x^{t-1})$	$\dfrac{-\alpha(z+\gamma(z-1))}{(z-1)(z-\beta)}$
Proximal methods	$y^{t+1} \in \arg\min_y f(y) + \frac{1}{2\alpha}\|y - y^t\|^2$	$\dfrac{-\alpha z}{z-1}$

Consensus

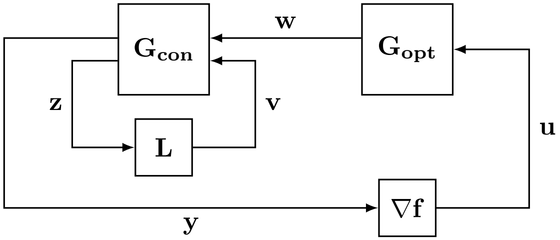


$$\begin{bmatrix} y_i \\ z_i \end{bmatrix} = G_{\text{con}} \begin{bmatrix} w_i \\ v_i \end{bmatrix}$$

$$v_i = \sum_{j=1}^{n} a_{ij} \left(z_i - z_j \right)$$

	Algorithm	$\widehat{G}_{\text{con}}(z)$
First-order estimator	$\mathbf{x}^{t+1} = \mathbf{L}(\mathbf{w}^t - \mathbf{x}^t)$ $\mathbf{y}^t = \mathbf{w}^t - \mathbf{x}^t$	$\begin{bmatrix} 1 & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix}$
Second-order estimator	$\mathbf{x}^{t+1} = 2\mathbf{x}^t - \mathbf{x}^{t-1} + \mathbf{L}\mathbf{y}^t$ $\mathbf{y}^t = \mathbf{w}^t - \frac{1}{2}(2\mathbf{x}^t - \mathbf{x}^{t-1})$	$\begin{bmatrix} 1 & \frac{\frac{1}{2}-z}{(z-1)^2} \\ 1 & \frac{\frac{1}{2}-z}{(z-1)^2} \end{bmatrix}$

A universal decomposition



- $\mathbf{G_{opt}}$ is a centralized optimization method
- $\mathbf{G_{con}}$ 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

$$\widehat{G}_{\mathrm{con}}(z) = \begin{bmatrix} 1 & \frac{\frac{1}{2}-z}{(z-1)^2} \\ 1 & \frac{\frac{1}{2}-z}{(z-1)^2} \end{bmatrix}$$

EXTRA

$$\widehat{G}_{\mathrm{con}}(z) = \begin{bmatrix} 1 & \frac{-\frac{1}{2}z^2}{(z-1)^2} \\ 1 & \frac{\frac{1}{2}-z}{(z-1)^2} \end{bmatrix}$$

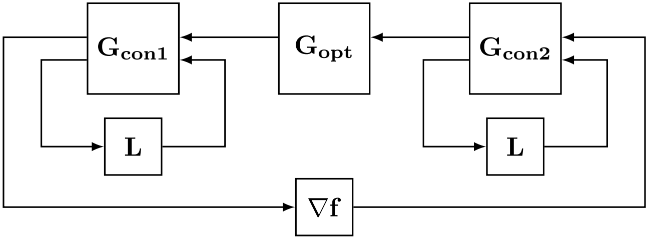
NIDS / ExDiff

$$\widehat{G}_{\mathrm{con}}(z) = \begin{bmatrix} 1 & \frac{-z(z+\beta-1)}{(z-1)^2} \\ 1 & \frac{1-(1+\beta)z}{(z-1)^2} \end{bmatrix}$$

SVL

In all cases, $\widehat{G}_{\mathrm{opt}}(z) = \frac{-\alpha}{z-1}$ (gradient descent).

Factored form



- $\mathbf{G_{opt}}$ is a centralized optimization method
- $\mathbf{G_{con1}}$ and $\mathbf{G_{con2}}$ 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

$$\widehat{G}_{\text{con1}}(z),\, \widehat{G}_{\text{con2}}(z) = \begin{bmatrix} 1 & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & \frac{-z}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix}$$

- DIGing uses the estimator on the left for both factors
- $\mathcal{AB}$ uses one of each estimator
- AugDGM uses the estimator on the right for both factors

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

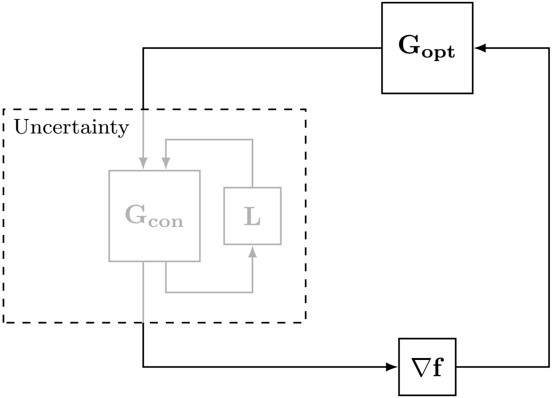
Accelerated algorithms that factor

$$\widehat{G}_{\text{con1}}(z) = \begin{bmatrix} 1 & \frac{1}{\alpha}\widehat{G}_{\text{opt}}(z) \\ 1 & \frac{1}{\alpha}\widehat{G}_{\text{opt}}(z) \end{bmatrix} \quad \text{and} \quad \widehat{G}_{\text{con2}}(z) = \begin{bmatrix} 1 & \frac{-1}{z-1} \\ 1 & \frac{-1}{z-1} \end{bmatrix}$$

- $\mathcal{ABN}$ uses $\gamma = \beta$ (Nesterov's method)
- $\mathcal{AB}m$ uses $\gamma = 0$ (heavy-ball method)

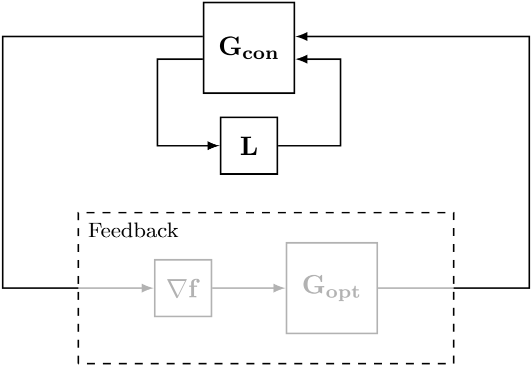
In all cases, $\widehat{G}_{\text{opt}}(z) = \frac{-\alpha(z+\gamma(z-1))}{(z-1)(z-\beta)}$ (accelerated method).

Interpretation: robust optimization



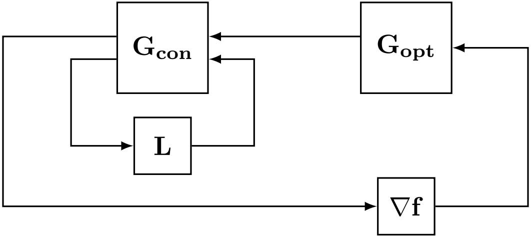
The optimization method must be robust to dynamic uncertainty.

Interpretation: consensus with feedback



The consensus estimator must be stable under nonlinear feedback.

Summary



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?