Push Sum Distributed Dual Averaging for Convex Optimization

Konstantinos I. Tsianos, PhD Candidate
Department of Electrical and Computer Engineering
Computer Networks Lab

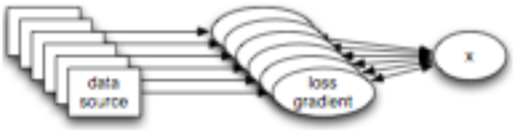
We generate a lot of data…

- 90 trillion emails sent in 2009
- 126 million blogs
- 4 billion photos in Flickr in 2009
- 2.5 billion photos uploaded each month on Facebook
- …

More data than a single machine can handle!

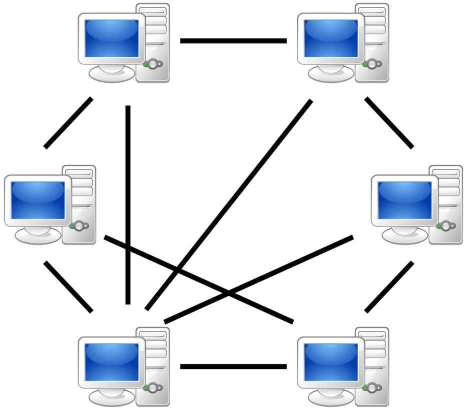
We turn to Distributed Computing

- Many (well structured) paradigms e.g.:
  - Clusters – MapReduce, MPI
  - GPUs – Cuda



[Langford09]

- **Peer-to-Peer type architectures**
  - Scalability
  - Robustness to failures
  - Simplicity of implementation
  - Less structured



Distributed Convex Optimization

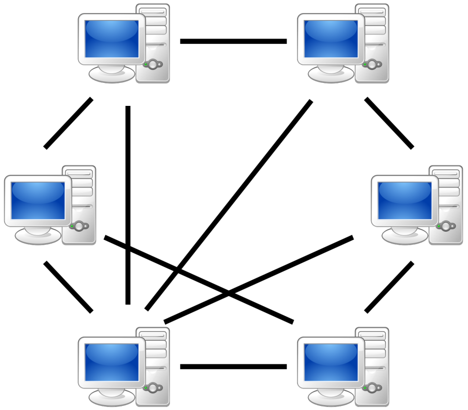
- Network G=(V,E) of n nodes
- Each node holds a convex function

$$f_i(x),\ \ x \in \mathcal{R}^d$$

- Minimize $f(x) = \sum_{i=1}^{n} f_i(x)$

  subject to convex constraints

$$x \in \mathcal{X}$$



Example: Linear Classification

- (Lots of) Labeled Data: $D = \{(a_j, y_j)\}_{j=1}^{m}$
- Train Linear Classifier (learn x):

$$\hat{y} = c(a|x) = sign(\langle a, x \rangle)$$

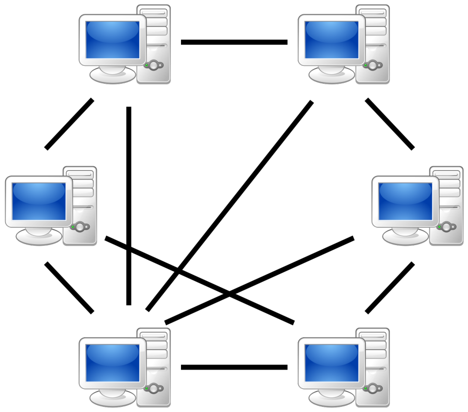
- By minimizing the hinge loss function

$$f(x) = \sum_{j=1}^{m} f_j(x) = \sum_{j=1}^{m} [1 - y_j \langle x, a_j \rangle]_+$$

Consensus Based Distributed Optimization

- Interleave communication with computation
- Local gradient steps optimize $f_i(x)$
- A consensus protocol brings the estimates to an agreement
- Many recent algorithms

[**Duchi11**, Ram11, Boyd10, Nedic09, Johansson09,…]



Dual Averaging [Nesterov09]

- Minimize $f(x), \; x \in \mathcal{X}$ by repeating

$$z(t+1) = z(t) + g(t), \quad g(t) = \frac{\partial f(x)}{\partial x}\Big|_{x=x(t)}$$

$$x(t+1) = argmin_{x \in \mathcal{X}} \left\{ \langle z(t+1), x \rangle + \frac{1}{a(t)} \psi(x) \right\}$$

$\psi(x)$ Strongly convex function

$a(t)$ Step size sequence

Dual Averaging [Nesterov09]

- Minimize $f(x), \;\; x \in \mathcal{X}$ by repeating

$$z(t+1) = z(t) + g(t), \quad g(t) = \frac{\partial f(x)}{\partial x}\Big|_{x=x(t)}$$

$$x(t+1) = argmin_{x \in \mathcal{X}} \left\{ \langle z(t+1), x \rangle + \frac{1}{a(t)} \psi(x) \right\}$$

$\psi(x)$ Strongly convex function

$a(t)$ Step size sequence

Convergence:

$$\hat{x}(T) = \frac{1}{T} \sum_{t=1}^{T} x(t) \to x_{opt}$$

Dual Averaging [Nesterov09]

- Minimize $f(x),\ x \in \mathcal{X}$ by repeating

$$z(t+1) = z(t) + g(t), \quad g(t) = \frac{\partial f(x)}{\partial x}\Big|_{x=x(t)}$$

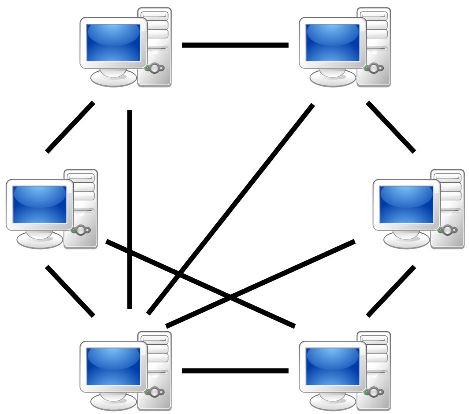
$$x(t+1) = argmin_{x \in \mathcal{X}} \left\{ \langle z(t+1), x \rangle + \frac{1}{a(t)} \psi(x) \right\}$$

  - For $\mathcal{X} = \mathcal{R}^d,\ \psi(x) = \frac{x^T x}{2}$ reduces to

$$x(t+1) = a(t) z(t+1)$$

Distributed Dual Averaging (DDA) [Ducchi11]

- Minimize $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x),\ \ x \in \mathcal{X}$
- By repeating at each node



$$z_i(t) = \sum_{j=1}^{n} p_{ij} z_j(t-1) + g_i(t-1)$$

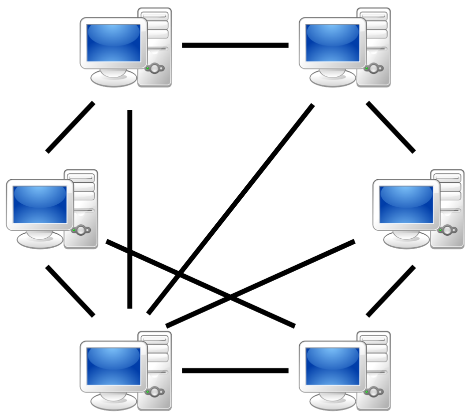
$$x_i(t) = argmin_{x\in\mathcal{X}}\left\{\langle z_i(t), x\rangle + \frac{1}{a(t)}\psi(x)\right\}$$

$\psi(x)$ Strongly convex function

$a(t)$ Step size sequence

Distributed Dual Averaging (DDA) [Ducchi11]

- Minimize $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \;\; x \in \mathcal{X}$



- By repeating at each node

$$z_i(t) = \sum_{j=1}^{n} p_{ij} z_j(t-1) + g_i(t-1)$$

$$x_i(t) = argmin_{x\in\mathcal{X}}\left\{\langle z_i(t), x\rangle + \frac{1}{a(t)}\psi(x)\right\}$$

With P doubly stochastic: $\hat{x}_i(T) = \frac{1}{T}\sum_{t=1}^{T} x_i(t) \rightarrow x_{opt}$

Distributed Consensus

- Network G=(V,E) of n nodes
- Initial value vector: $x(0)$
- Iterate: $x(t+1) = Px(t) = P^t x(0)$
- Trying to reach consensus: $x(t) \to c\mathbf{1}$
- Convergence: $[P^t]_{i,:} \to \pi^T, P\mathbf{1} = \mathbf{1}$

  or $[P^t]_{i,:} \to \pi_i \mathbf{1}^T, \mathbf{1}^T P = \mathbf{1}^T$

Distributed Consensus

- Network G=(V,E) of n nodes
- Initial value vector: $x(0)$
- Iterate: $x(t+1) = Px(t) = P^t x(0)$
- Trying to reach consensus: $x(t) \rightarrow c\mathbf{1}$
- Convergence: $[P^t]_{i,:} \rightarrow \pi^T, P\mathbf{1} = \mathbf{1}$

  or $[P^t]_{i,:} \rightarrow \pi_i \mathbf{1}^T, \mathbf{1}^T P = \mathbf{1}^T$

Distributed Consensus

- Network G=(V,E) of n nodes
- Initial value vector: $x(0)$
- Iterate: $x(t+1) = Px(t) = P^t x(0)$
- Trying to reach consensus: $x(t) \to c\mathbf{1}$
- Convergence: $[P^t]_{i,:} \to \pi^T, P\mathbf{1} = \mathbf{1}$

or $[P^t]_{i,:} \to \pi_i \mathbf{1}^T, \mathbf{1}^T P = \mathbf{1}^T$

Optimization Bias

$$z_i(t) = \sum_{j=1}^{n} p_{ij} z_j(t-1) + g_i(t-1)$$

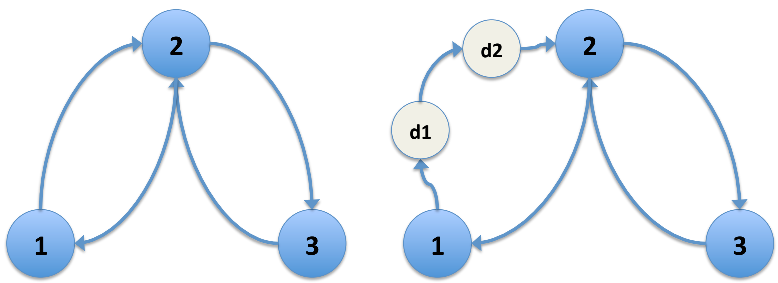
After back-substituting in the recursion:

$$z_i(t) = \sum_{r=1}^{t} \sum_{j=1}^{n} [P^{t-r}]_{ij} g_j(r-1)$$

If P does not have a uniform stationary distribution, the z variables are weighted unequally. Minimizes

$$f(x) = \sum_{i=1}^{n} \pi_i f_i(x)$$ instead of $$f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

Fixed Communication Delays



$$x_i(t) = \sum_{j=1}^{n+b} q_{ij} x_j(t-1)$$

- Maximum delay per edge is B
- Total amount of delay is b
- Matrix with delays Q constructed from P
  - Q is stochastic
  - Q corresponds to a non-reversible Markov chain

General Consensus Matrices and Semantics

- Restrict to Average Consensus: $\pi = \frac{1}{n}\mathbf{1}$
- Consensus protocol with fixed delays Q(P) is not doubly stochastic even if P is [Tsianos11]
- Not all networks admit a doubly stochastic consensus protocol P [Gharesifard10]

- Row stochastic P: Receiver forms convex combination of incoming messages
- Column stochastic P: Receiver just sums incoming information

Time varying consensus matrices P(t)

- Model asynchronous communication and random delays
- Sparsify Communication
- Avoid “slow” node problem
- Receive unknown number of messages per iteration

- Still need averaging…

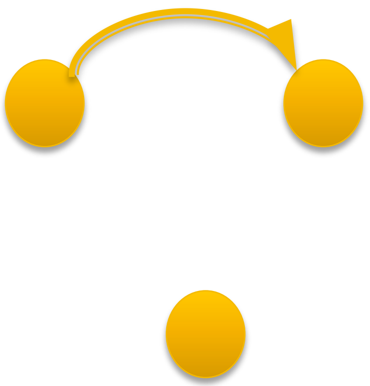
Time varying consensus matrices P(t)

- Model asynchronous communication and random delays
- **Sparsify Communication**
- Avoid “slow” node problem
- Receive unknown number of messages per iteration

- Still need averaging…

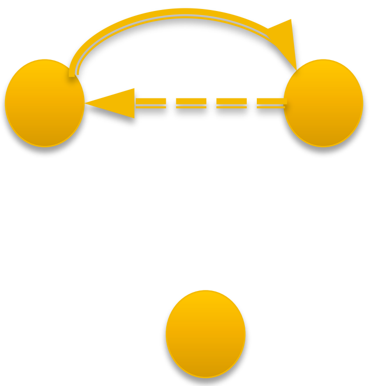
One Directional vs Bi-directional Communication

- Bi-directional communication causes deadlocks!



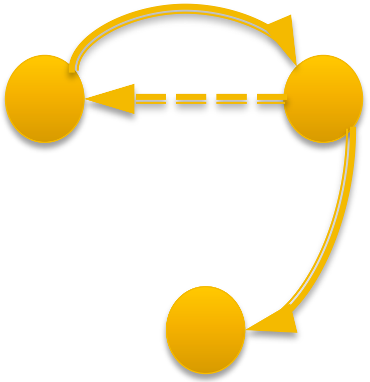
One Directional vs Bi-directional Communication

- Bi-directional communication causes deadlocks!



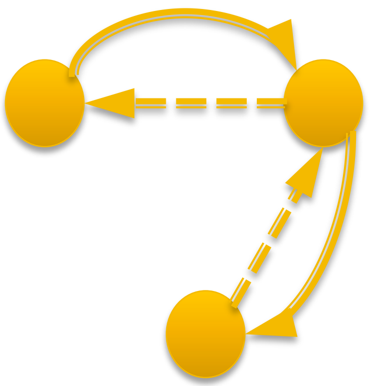
One Directional vs Bi-directional Communication

- Bi-directional communication causes deadlocks!



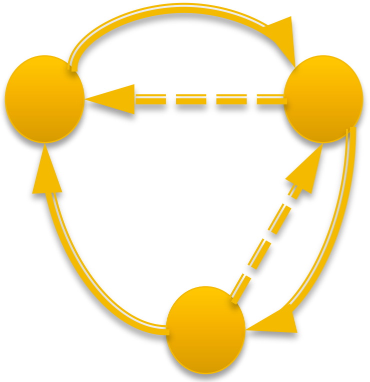
One Directional vs Bi-directional Communication

- Bi-directional communication causes deadlocks!



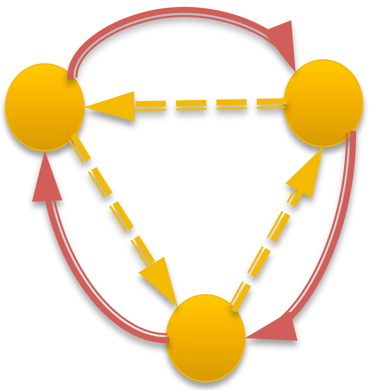
One Directional vs Bi-directional Communication

- Bi-directional communication causes deadlocks!



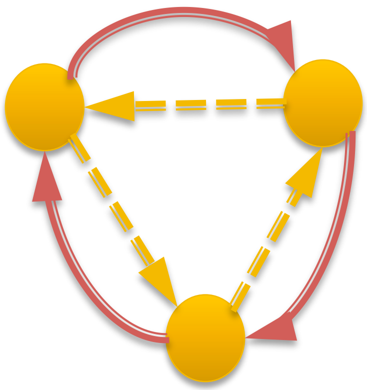
One Directional vs Bi-directional Communication

- Bi-directional communication causes deadlocks!



One Directional vs Bi-directional Communication

- Bi-directional communication causes deadlocks!



- One directional communication with doubly stochastic P(t) requires coordination
- Row stochastic P(t): Convergence to the average in expectation [Aysal08]

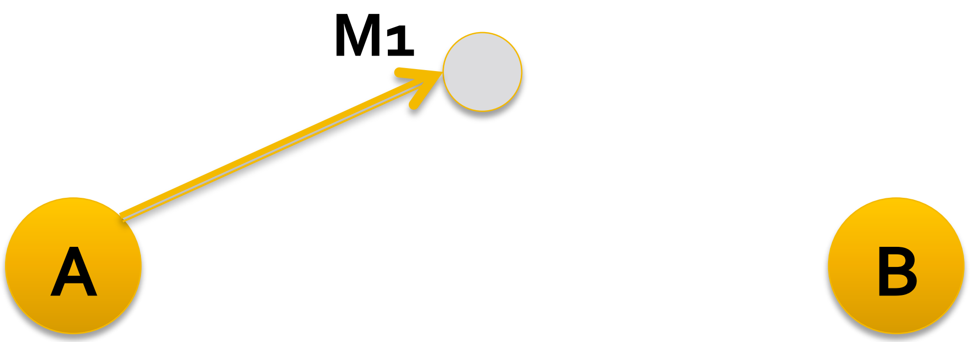
Time varying consensus matrices P(t)

- Model asynchronous communication and random delays
- Sparsify Communication
- Avoid “slow” node problem
- **Receive unknown number of messages per iteration**

- Still need averaging…

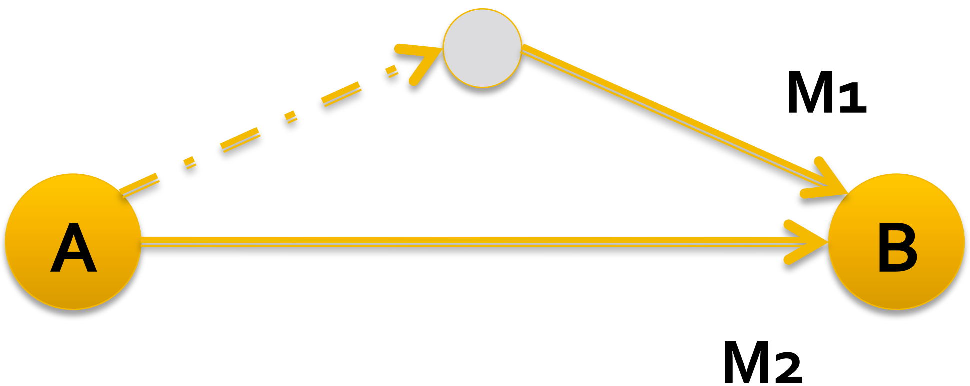
Example: Receiving Multiple Messages due to Random Delay

- Time $t = 1$
  - Node A sends a message M1 to B with delay 1



Example: Receiving Multiple Messages due to Random Delay

- Time $t = 2$
  - Node A sends a message M2 to B with no delay



- M1 and M2 are delivered at the same time!
  - Not captured by $x_i(t+1) = \sum_{j=1}^{n} p_{ij} x_j(t - b_{ij}(t))$

Push-Sum Consensus

- Column stochastic P
- One directional
- Receiver simply adds incoming messages
  - Can receive varying number of messages at each iteration

$$w(0) = \mathbf{1} \qquad s(0) = x(0)$$
$$w(t) = Pw(t-1)$$
$$s(t) = Ps(t-1)$$
$$x(t) = \frac{s(t)}{w(t)} \rightarrow \frac{\mathbf{1}^T x(0)}{n}$$

- Converges to the true average for fixed protocols P and time varying protocols P(t) [Kempe03,Benezit10]
  - Do not need to know the stationary distribution of P in advance

Push Sum Distributed Dual Averaging (PS-DDA)

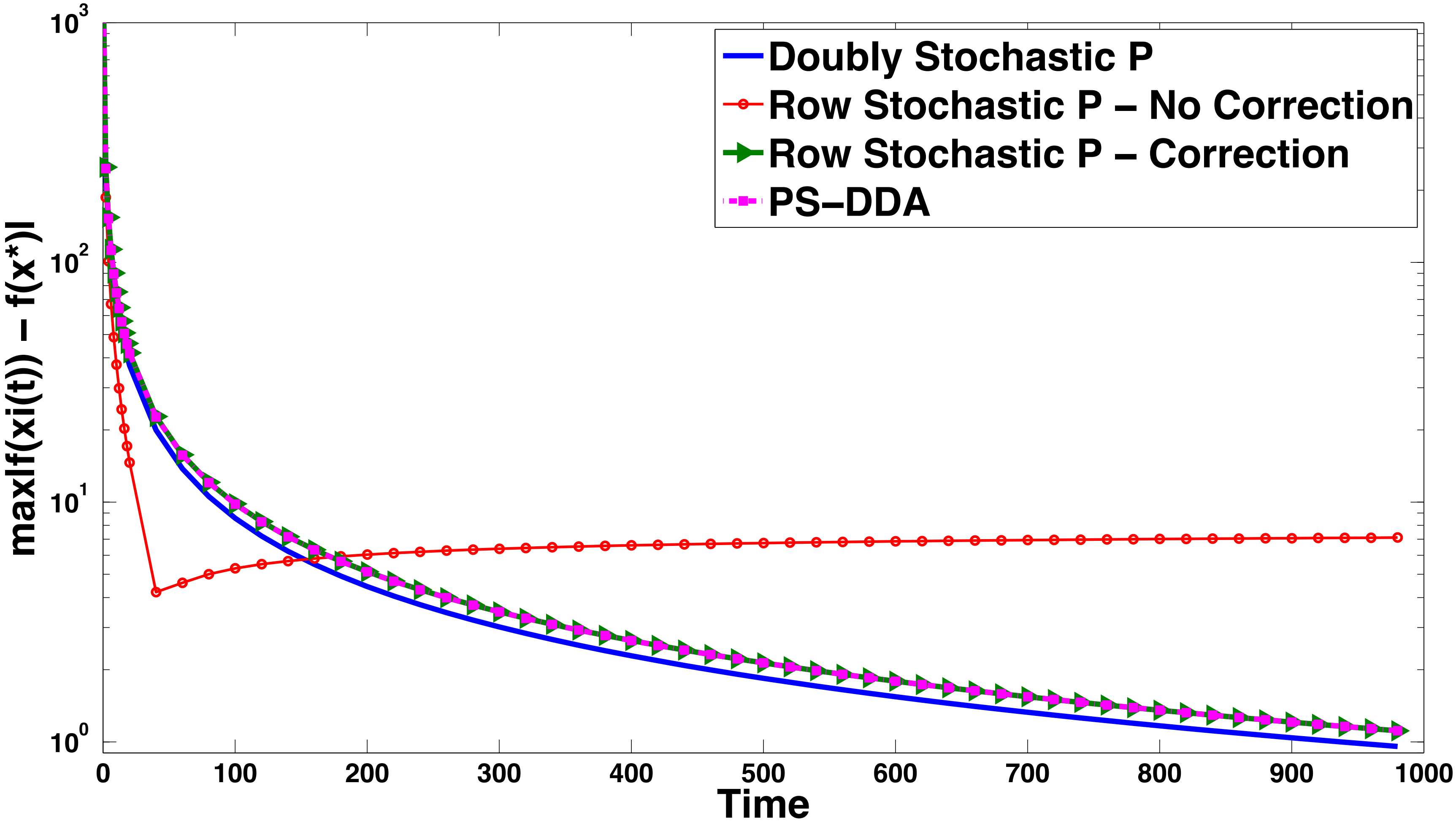
$$w_i(t+1) = \sum_{j=1}^{n} p_{ij} w_j(t)$$

$$z_i(t+1) = \sum_{j=1}^{n} p_{ij} z_j(t) + g_i(t)$$

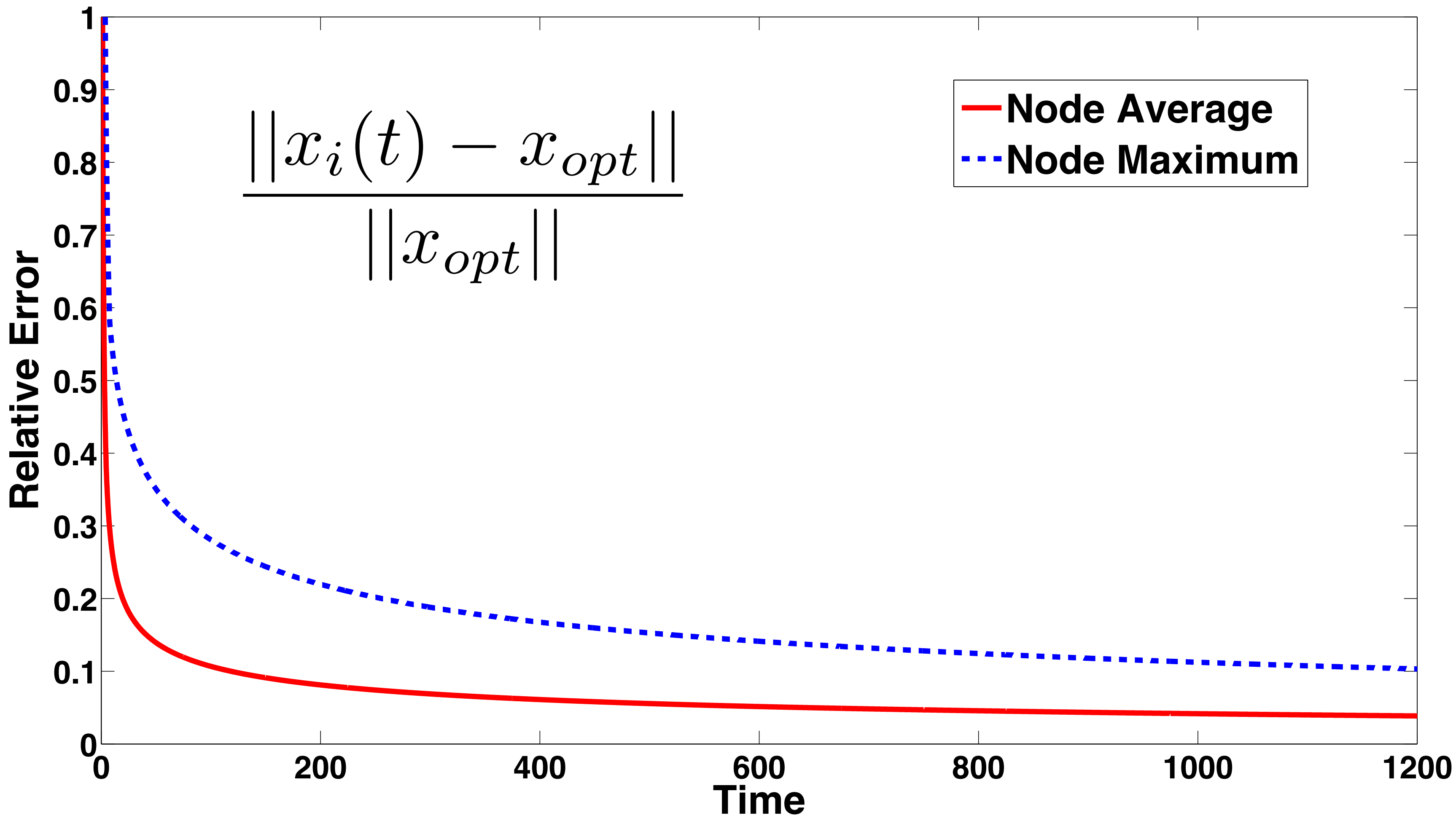
$$x_i(t+1) = argmin_{x \in \mathcal{X}} \left\{ \left\langle \frac{z_i(t+1)}{w_i(t+1)}, x \right\rangle + \frac{1}{a(t)} \psi(x) \right\}$$

- [Tsianos12] Synchronous PS-DDA converges to the unbiased optimum at a rate $O(T^{-0.5})$ same as standard DDA
  - Convergence can be generalized to asynchronous communication and communication with delays

Example: Minimize a sum of quadratics (Simulation)



Minimize 150.000 quadratics with 15 nodes on an MPI cluster



Summary

- Consensus Based Distributed Optimization – Distributed Dual Averaging
- Role of Consensus Protocol
  - Optimization Bias, Deadlocks
  - Row vs Column stochastic protocols
  - Push Sum Consensus
- Push Sum Distributed Dual Averaging
  - Convergence
  - Communication Delays

Future Work

- How to combine information to promote optimization?
  - Adapt the consensus protocol?
- What is the effect of having a sum of very different local function?
- What is gained when using second order information?
- Can we converge to a local solution with a non-convex objective?
- When do we scale with the network size n?

Consensus with Fixed Delays

- Convergence to Consensus [Tsianos11]

$$||Q^t(i,\cdot) - \pi^T||_{TV}^2 \leq \frac{(\lambda_2(U))^t}{4\pi_i}$$

$$U = \frac{Q_l + \hat{Q}_l}{2}, \quad Q_l = \frac{1}{2}(I + Q)$$

- Effect of maximum delay B on convergence rate [New Result]

$$\lambda_2(P) \leq 1 - \frac{1}{K} \Rightarrow \lambda_2(U) \leq 1 - \frac{1}{ZK}, \quad Z = O(B^2)$$

Distributed Dual Averaging with Fixed Delays

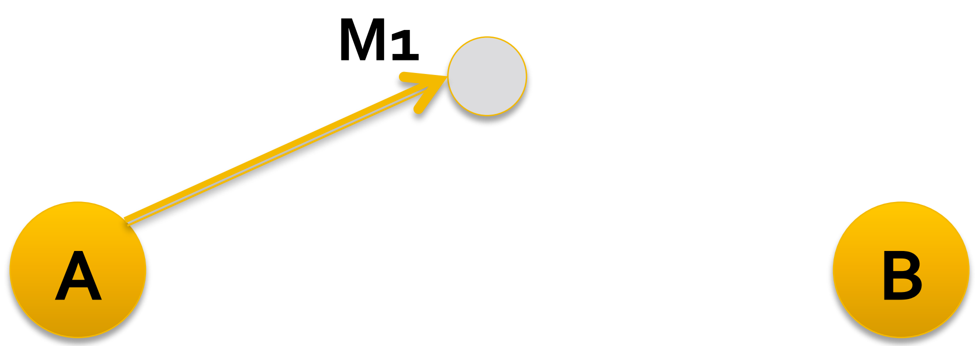
- [Theorem 2, Tsianos11] *DDA converges to the optimal solution for any row stochastic protocol P and thus for any fixed edge delays.*

- [Theorem 3, Tsianos11] *If P is doubly stochastic*

$$error(t) \leq 2RL\frac{n+b}{n}\sqrt{13 + \frac{6\lambda_2(Q)\sqrt{n+b}}{1-\sqrt{\lambda_2(Q)}}}\frac{1}{\sqrt{t}}$$

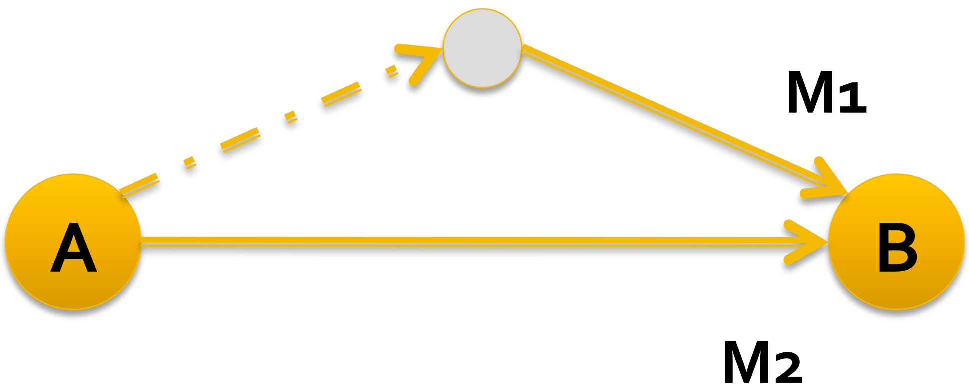
Example: Receiving Multiple Messages due to Random Delay

- Time $t = 1$
  - Node A sends a message M1 to B with delay 1



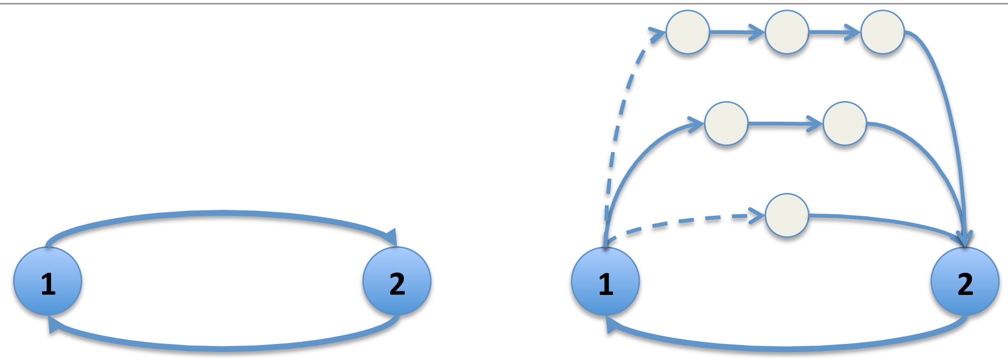
Example: Receiving Multiple Messages due to Random Delay

- Time $t = 2$
  - Node A sends a message M2 to B with no delay



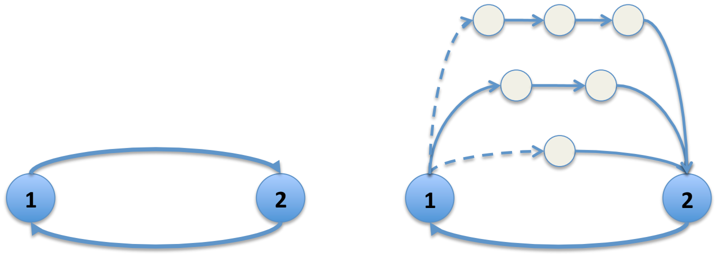
- M1 and M2 are delivered at the same time!
  - Not captured by $x_i(t+1) = \sum_{j=1}^{n} p_{ij} x_j(t - b_{ij}(t))$

Random Delay Model



- Add $b = \dfrac{mB(B+1)}{2}$ delay nodes
- At each iteration chose a delay path per edge
- Choose an adjacency matrix $A(t)$ out of $\{A^1, \ldots, A^{B^m}\}$
- Construct the consensus matrix $Q(t)$
- Next consensus iteration is $x_i(t+1) = \sum_{i=1}^{n+b} q_{ij}(t) x_j(t)$

Convergence with Random Delays



- "Row stochastic" Q(t)
  - Q(t) contains zero rows!
  - Convergence to consensus for compute nodes for any row stochastic P [New Result]

- Column stochastic Q(t)
  - Push-sum converges to the average consensus for any column stochastic P(t)
  - Convergence rate is exponential but pessimistic

Scalability with Network Size

- Current DDA bound does not scale with n

$$error(t) = O\left(\frac{\log(t\sqrt{n})}{\sqrt{t}}\right)$$

- Is it possible to converge faster with more processors?
- Is it beneficial to sparsify communication?

[New Result: Not always!]

PS-DDA Convergence Rate Bound

$$f(\hat{x}_i(T)) - f(x_{opt}) \leq 2RL\sqrt{1 + \frac{8+4n}{c\sqrt{\pi^*}(1-\sqrt{\lambda_2})}}\frac{1}{\sqrt{T}}$$