Distributed Formation of Balanced and Bistochastic Weighted Digraphs in Multi-Agent Systems

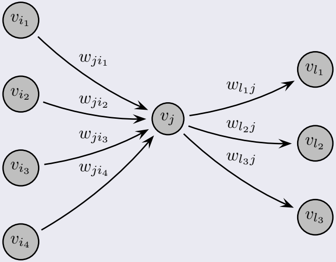
T. Charalambous[‡] and C. N. Hadjicostis[⋆]

[‡] Department of Automatic Control, Royal Institute of Technology (KTH)

[⋆] Department of Electrical and Computer Engineering, University of Cyprus

European Control Conference (ECC '13)
Zurich, Switzerland, July 2013

Distributed Formation of Balanced and Bistochastic Weighted Digraphs in Multi-Agent Systems



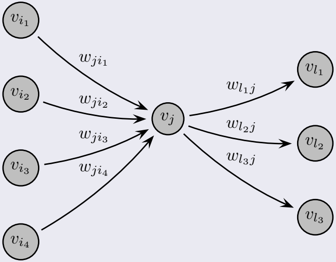
Weight-balanced digraph:

Sum of weights on incoming links = Sum of weights on outgoing links

Bistochastic (doubly stochastic) digraph:

Sum of weights on incoming links = Sum of weights on outgoing links = 1

Distributed Formation of Balanced and Bistochastic Weighted Digraphs in Multi-Agent Systems



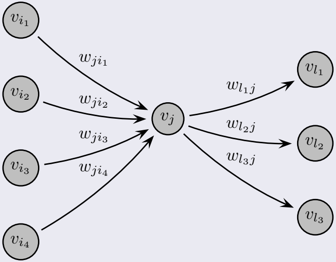
Weight-balanced digraph:

Sum of weights on incoming links = Sum of weights on outgoing links

Bistochastic (doubly stochastic) digraph:

Sum of weights on incoming links = Sum of weights on outgoing links = 1

Distributed Formation of Balanced and Bistochastic Weighted Digraphs in Multi-Agent Systems



Weight-balanced digraph:

Sum of weights on incoming links = Sum of weights on outgoing links

Bistochastic (doubly stochastic) digraph:

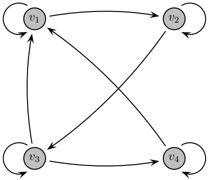
Sum of weights on incoming links = Sum of weights on outgoing links = 1

Motivation

1. **Weight-balanced matrix formation**
   - Synchronization
   - Average consensus via linear iterations for continuous-time systems (special case of synchronization without dynamics)
   - Applications where weight balance plays a key role:
     - Traffic-flow problems captured by n junctions and m one-way streets
     - Stable flocking of agents with significant inertial effects
     - Pinning control, optoelectronics, biology, ...
   - Related to weights that form a bistochastic matrix

2. **Bistochastic matrix formation**
   - Average consensus via linear iterations in discrete-time systems - applications in multicomponent systems where one is interested in distributively averaging measurements, e.g., sensor networks, environmental monitoring

- Distributed systems conveniently captured by digraphs
  1. Components represented by vertices (nodes)
  2. Communication and sensing links represented by edges



- Consider a network with nodes $(v_1, v_2, \ldots, v_N)$
  E.g., sensors, robots, unmanned vehicles, resources, etc.

- Nodes can receive information according to (possibly directed) communication links

- Each node v_j has some initial value $x_j[0]$ (could be belief, position, velocity, etc.)

Consensus and Average Consensus

- Typical objective: Calculate functions of initial values in a distributed manner (e.g., $\max_\ell\{x_\ell[0]\}$, $\sum_\ell x_\ell^2[0]$, etc.)
- Consensus: All nodes calculate (in a distributed manner, each time using only local information) *same function* of initial values $x_1[0]$, $x_2[0]$, $\ldots$, $x_N[0]$
- Average Consensus: All nodes calculate (in a distributed manner) the *average* $\overline{x} \equiv \frac{1}{N}\sum_{\ell=1}^{N} x_\ell[0]$ (where N is the number of nodes)
- Possible centralized strategy: Route all values to a single entity (leading node) who then determines the function value (e.g., average) and routes it back to the nodes
- Average serves as primitive for estimation, inference and diagnosis (easily adjusted to arbitrary linear functions)

Outline

- Notation and mathematical preliminaries
- Weight-balanced matrix formation
- Bistochastic matrix formation
- Comparisons
- Concluding remarks and future directions

Graph Notation

- Digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - Nodes (system components) $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$
  - Edges (directed communication links) $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ where $(v_j, v_i) \in \mathcal{E}$ iff node v_j can receive information from node v_i
  - In-neighbors $\mathcal{N}_j^- = \{v_i \mid (v_j, v_i) \in \mathcal{E}\}$; in-degree $\mathcal{D}_j^- = |\mathcal{N}_j^-|$
  - Out-neighbors $\mathcal{N}_j^+ = \{v_l \mid (v_l, v_j) \in \mathcal{E}\}$; out-degree $\mathcal{D}_j^+ = |\mathcal{N}_j^+|$
- Adjacency matrix A: $A(j,i) = 1$ if $(v_j, v_i) \in \mathcal{E}$; $A(j,i) = 0$ otherwise
- Undirected graph: $(v_j, v_i) \in \mathcal{E}$ iff $(v_i, v_j) \in \mathcal{E}$ (bidirectional links)
  In undirected graphs, we have (for each node j)
  $\mathcal{N}_j^+ = \mathcal{N}_j^-$ and $\mathcal{D}_j^+ = \mathcal{D}_j^- = \mathcal{D}_j$; also, $A = A^T$
- (Strongly) connected (di)graph if for any $i, j \in \mathcal{V}$, $j \neq i$, there exists a (directed) path connecting them, i.e.,

$$v_i = v_{i_0} \to v_{i_1},\ v_{i_1} \to v_{i_2},\ \ldots,\ v_{i_{t-1}} \to v_{i_t} = v_j$$

Weight-Balanced Matrix Formation

The Algorithm (1/2)

- Setting: Nodes distributively adjust the weights of their outgoing links such that the digraph asymptotically becomes weight-balanced; they observe but cannot set the weights of their incoming links
- Each node v_j initialize the weights of all of its outgoing links to unity, i.e., $w_{lj}[0] = 1, \forall v_l \in \mathcal{N}_j^+$ (different initial weights also possible)
- Nodes enter an iterative stage where node v_j performs the following steps:
  1. It computes its weight imbalance defined by
  $$x_j[k] \triangleq S_j^-[k] - S_j^+[k] \ ,$$
  where $S_j^- = \sum_{v_i \in \mathcal{N}_j^-} w_{ji}$ and $S_j^+ = \sum_{v_l \in \mathcal{N}_j^+} w_{lj}$
  2. If $x_j[k]$ is positive (resp. negative), all the weights of its outgoing links are increased (resp. decreased) by an equal amount and proportionally to $x_j[k]$, specifically, $\forall v_l \in \mathcal{N}_j^+$,
  $$w_{lj}[k+1] = w_{lj}[k] + \beta_j \left(\frac{S_j^-[k]}{\mathcal{D}_j^+} - w_{lj}[k] \right) , \quad \beta_j \in (0,1)$$

- Intuition: we compare $S_j^-[k]$ with $S_j^+[k] = \mathcal{D}_j^+ w_{lj}[k]$. If $S_j^+[k] > S_j^-[k]$ (resp. $S_j^+[k] < S_j^-[k]$), then the algorithm reduces (resp. increases) the weights on the outgoing links

Proposition 1

If a digraph is strongly connected, **the weight balancing algorithm** asymptotically reaches a steady state weight matrix W^* that forms a weight-balanced digraph, with geometric convergence rate equal to $R_\infty(P) = -\ln \delta(P)$, where

$$P_{ji} \triangleq \begin{cases} 1-\beta_j, & \text{if } i=j, \\ \beta_j/\mathcal{D}_j^+, & \text{if } v_i \in \mathcal{N}_j^-, \end{cases}$$

and $\delta(P) \triangleq \max\{|\lambda| : \lambda \in \sigma(P)), \lambda \neq 1\}$

Weight-Balanced Matrix Formation

Sketch of the Proof (1/2)

- Observation: $w_{l'j} = w_{lj}, \forall v_{l'}, v_l \in \mathcal{N}_j^+$ (because they are equal at initialization and they are updated in the same fashion)
- Hence, we denote the weight on any outgoing link of node v_j as w_j
- We define $w = (w_1 \;\; w_2 \;\; \ldots \;\; w_n)^T$ with $w_j = w_{lj}$ ($v_l \in \mathcal{N}_j^+$)
- Then, the evolution of the weights in matrix form is as follows

$$w[k+1] = Pw[k], w[0] = w_0 \;, \tag{1}$$

where

$$P_{ji} \triangleq \begin{cases} 1 - \beta_j, & \text{if } i = j, \\ \beta_j / \mathcal{D}_j^+, & \text{if } v_i \in \mathcal{N}_j^- \end{cases} \tag{2}$$

- The above update equation implies that the weights remain nonnegative during the execution of the algorithm

- Matrix P can be written as $P = I - B + BD^{-1}A$, where I is the identity matrix, $B = diag(\beta_j)$, $D = diag(\mathcal{D}_j^+)$ and A is the adjacency matrix
- Note: $\bar{P} \triangleq I - B + AD^{-1}B$ is column stochastic and therefore $\rho(\bar{P}) = 1$
- With simple algebraic manipulation

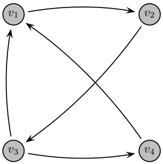
$$\rho(\bar{P}) = \rho(\bar{P}B^{-1}DD^{-1}B) = \rho(D^{-1}B\bar{P}B^{-1}D) = \rho(P) = 1 .$$

- Since the digraph is strongly connected for $0 < \beta_j < 1$, $\forall v_j \in \mathcal{V}$, and all the main diagonal entries are positive, P is primitive
- Hence, iteration (1) has a geometric convergence rate

$$R_\infty(P) = -\ln \delta(P) \; ,$$

where $\delta(P)$ is the second largest of the moduli of the eigenvalues of P

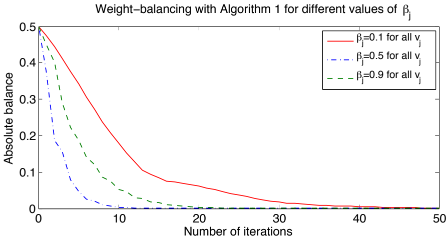
- Example borrowed by [B.Gharesifard & J.Cortés, 2010]



- Concerned with the absolute balance defined as

$$\varepsilon[k] = \sum_{j=1}^{n} |x_j[k]|$$

- If weight balance is achieved, then $\varepsilon[k] = 0$ and $x_j[k] = 0, \forall v_j \in \mathcal{V}$

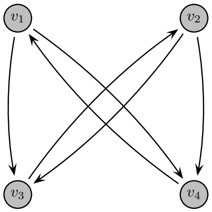


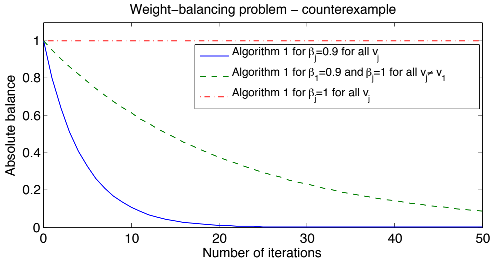
- Same $W^\star$ for $\beta_j = 0.1, 0.5, 0.9$

$$W^\star = \begin{bmatrix} 0 & 0 & 0.7143 & 0.7143 \\ 1.4286 & 0 & 0 & 0 \\ 0 & 1.4286 & 0 & 0 \\ 0 & 0 & 0.7143 & 0 \end{bmatrix}$$

B. Gharesifard and J. Cortés, “When does a digraph admit a doubly stochastic adjacency matrix?” in Proc. of American Control Conference, 2010.

- If $\beta_j = 1$, $\forall v_j \in \mathcal{V}$, then the weighted adjacency matrix P is not necessarily primitive
- Algorithm does not converge to weights that form a weight-balanced digraph





- For the case for which $\beta_j = 1$, $\forall v_j \in \mathcal{V}$ the matrix is not primitive and the algorithm does not converge, whereas for the other two cases it converges

- Necessary and sufficient conditions on P for asymptotic average consensus [Xiao & Boyd, 2004]
  1. P has a simple eigenvalue at 1 with left eigenvector $\mathbf{1}^T = [1\ 1\ \ldots\ 1]$ and right eigenvector $\mathbf{1} = [1\ 1\ \ldots\ 1]^T$
  2. All other eigenvalues of P have magnitude strictly smaller than 1
- As $k \to \infty$, $P^k \to \frac{1}{N}\mathbf{1}\mathbf{1}^T$ which implies that

$$\lim_{k\to\infty} x[k] = \frac{1}{N}\mathbf{1}\mathbf{1}^T x[0] = \left(\frac{\sum_{\ell=1}^{N} x_\ell[0]}{N}\right)\mathbf{1} \equiv \bar{x}\mathbf{1}$$

- Nonnegative $p_{ji} \implies P$ is primitive bistochastic

How to distributively obtain weights that form a bistochastic matrix?

L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging," Systems and Control Letters, Sept. 2004.

Bistochastic Matrix Formation
Intuition

- Extra requirement: maintain column stochasticity of the weighted adjacency matrix $W[k]$ for all times k
- Obtain a sequence of column stochastic matrices $W[0], W[1], \ldots, W[k]$ such that $\lim_{k\to\infty} W[k] = W$ is bistochastic and thus the iteration

$$x[k+1] = W[k]x[k], \quad x[0] = x_0 ,$$

  reaches average consensus asymptotically [A.Dominguez-Garcia & C.N.Hadjicostis, 2013]
- Digraphs that are weight-balanceable do not necessarily admit a doubly stochastic assignment [B.Gharesifard & J.Cortés, 2010]
- Any strongly connected graph is bistochasticable after adding enough self-loops [B.Gharesifard & J.Cortés, 2010]
- Thus, problem is overcome with the introduction of nonzero self weights (as long as graph is strongly connected)

A. Dominguez-Garcia and C. N. Hadjicostis, “Distributed matrix scaling and application to average consensus in directed graphs,” IEEE Transactions on Automatic Control, March 2013.

B. Gharesifard and J. Cortés, “When does a digraph admit a doubly stochastic adjacency matrix?” in Proc. of American Control Conference,2010.

- Each node v_j initializes the weights of all of its outgoing links to $w_{lj}[0] = 1/(1+\mathcal{D}_j^+)$, $\forall v_l \in \mathcal{N}_j^+$ (different initial weights also possible)
- Nodes enter an iterative stage where node v_j
  1. Chooses $\beta_j[k]$ as

$$\beta_j[k] = \begin{cases} \alpha_j \frac{1 - S_j^+[k]}{S_j^-[k] - S_j^+[k]}, & S_j^-[k] > S_j^+[k], \\ \alpha_j, & \text{otherwise}, \end{cases}$$

     where $\alpha_j \in (0,1)$
  2. Updates the weights of its outgoing links w_{lj}, $\forall v_l \in \mathcal{N}_j^+$

$$w_{lj}[k+1] = w_{lj}[k] + \beta_j[k] \left(\frac{S_j^-[k]}{\mathcal{D}_j^+} - w_{lj}[k] \right), \quad \beta_j[k] \in (0,1)$$

  3. Assigns $w_{jj} \geq 0$ so that the weighted adjacency matrix retains its column stochasticity, i.e.,

$$w_{jj}[k+1] = 1 - \sum_{l \in \mathcal{N}_j^+} w_{lj}[k+1], \ \forall v_j \in \mathcal{V}$$

Proposition 2

If a digraph is strongly connected, then the bistochastic matrix formation algorithm reaches a steady state weight matrix W^* that forms a bistochastic digraph; furthermore, the weights of all edges in the graph are nonzero

Proposition 3

If a digraph is strongly connected or is a collection of strongly connected digraphs, the algorithm with initial condition $w_{lj}[0] = \frac{1}{m(1+\mathcal{D}_j^+)}$, $\forall v_l \in \mathcal{N}_j^+, m \geq |\mathcal{V}|$, reaches a steady state weight matrix W^* that forms a bistochastic digraph, with geometric convergence rate equal to $R_\infty(P) = -\ln \delta(P)$, where

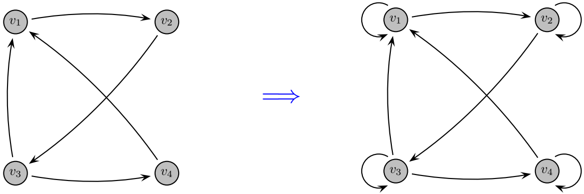
$$P_{ji}[k] \triangleq \begin{cases} 1 - \alpha_j \ , & \text{if } i = j, \\ \alpha_j / \mathcal{D}_j^+ \ , & \text{if } i \in \mathcal{N}_j^- . \end{cases}$$

Furthermore, the weights of all edges in the graph are nonzero

Bistochastic Matrix Formation

Illustrative Example (1/2)

- Same as before, with the difference being that self loops are introduced, i.e., self-weights are also updated



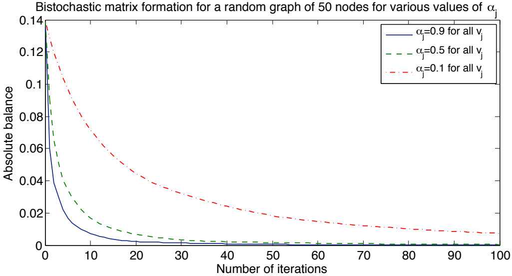
- The digraph above depicts the update of the self weight with the introduction of the self-loops at the nodes
- The adjacency matrix becomes bistochastible

Bistochastic Matrix Formation

Illustrative Example (2/2)

- Consider a random strongly connected graph consisting of 50 nodes
- Quantity of interest: the absolute balance, defined as

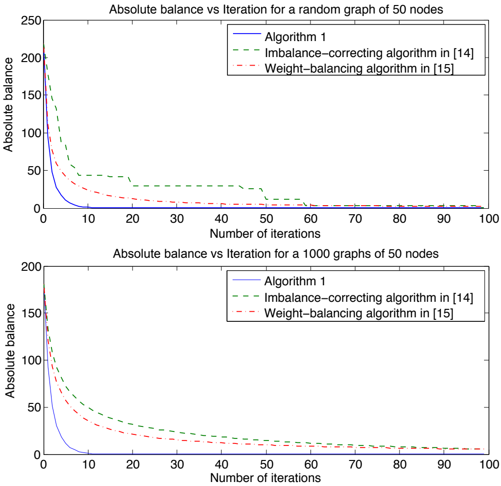
$$Ab[k] = \sum_{v_j \in \mathcal{V}} \left| 1 - \sum_{v_i \in \mathcal{N}_j^-} w_{ji}[k] \right|$$



- Asymptotically converges to a bistochastic adjacency matrix for different values of α_j

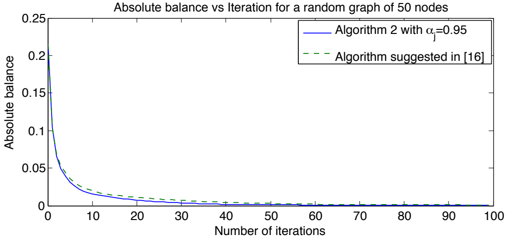
Comparisons

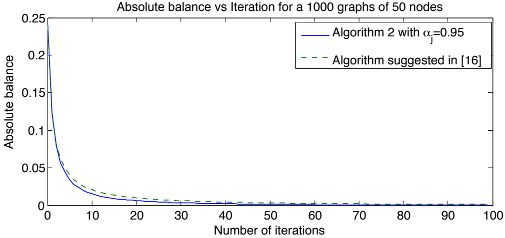
Weight-Balanced Matrix Formation



[14] B. Gharesifard and J. Cortés, “Distributed strategies for making a digraph weight-balanced,” in Proc. of Allerton Conference on Communication, Control, and Computing, 2009.

[15] C. N. Hadjicostis and A. Rikos, “Distributed strategies for balancing a weighted digraph,” in Proc. of the 20th Mediterranean Conference on Control Automation, 2012.





[16] A. Dominguez-Garcia and C. N. Hadjicostis, “Distributed matrix scaling and application to average consensus in directed graphs,” IEEE Transactions on Automatic Control, March 2013.

Concluding Remarks and Future Directions

Conclusions:

- Proposed a distributed algorithm for forming a **weight-balanced matrix**
- Proposed a distributed algorithm for forming a **bistochastic matrix**
- Weight-balanced matrix formation algorithm admits geometric convergence rates
- Bistochastic matrix formation algorithm *probably* admits geometric convergence rates
- Rate depends exclusively on the structure of the given digraph and constant parameters chosen by the nodes

Future work:

- Variations of distributed bistochastic matrix formation algorithms that provably admit geometric convergence rates
- Analysis of suggested algorithms in the presence of delays and changing topology

Thank You!

Questions?