Cycles in Random Graphs

Valery Van Kerrebroeck

Enzo Marinari, Guilhem Semerjian

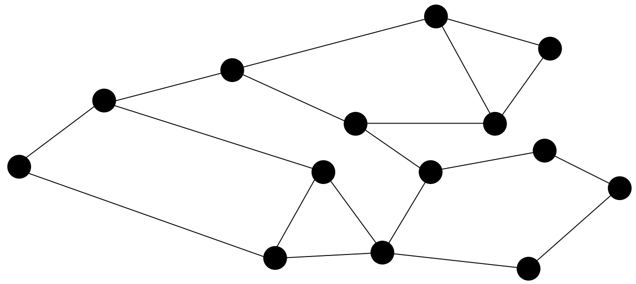
[Phys. Rev. E **75**, 066708 (2007)]

[J. Phys. Conf. Series **95**, 012014 (2008)]

Outline

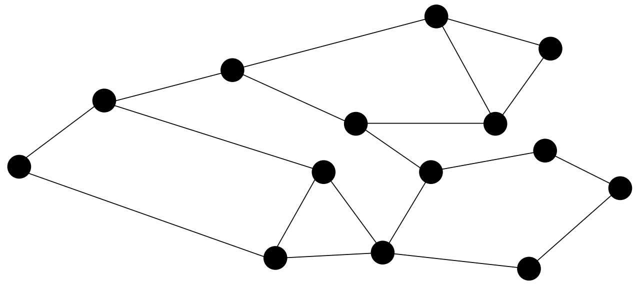
- Introduction
- Statistical Mechanics Approach
- Application 1: Finding Long Cycles
- Application 2: Vertex and Edge Ranking
- Conclusions and Future Perspectives

Definitions



Simple, Undirected Graph G(N,M)
has N vertices i and M edges $\{i, j\}$

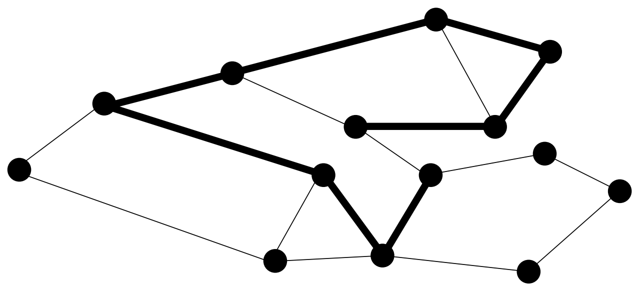
Definitions



Simple, Undirected Graph G(N,M) has N vertices i and M edges $\{i, j\}$

Walk of length L is a *sequence* $(i_0, i_1, \ldots, i_L)$ where each one of the vertices i_k is adjacent to i_{k+1} for all $k = 0, 1, \ldots, L - 1$

Definitions

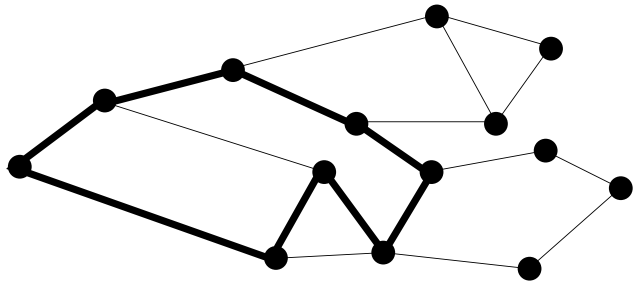


Simple, Undirected Graph G(N,M)
has N vertices i and M edges $\{i, j\}$

Walk of length L is a *sequence* $(i_0, i_1, \ldots, i_L)$ where each one of the vertices i_k is adjacent to i_{k+1} for all $k = 0, 1, \ldots, L-1$

Path of length L is a *non self-intersecting walk* passing through L edges of a graph.

Definitions



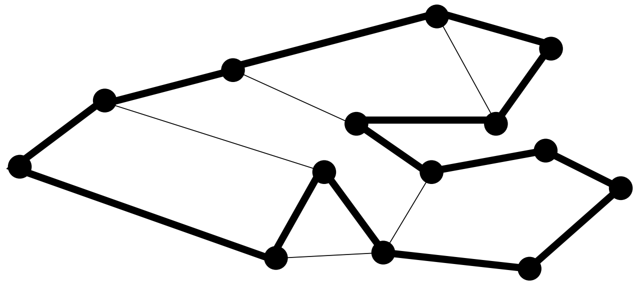
Simple, Undirected Graph G(N,M) has N vertices i and M edges $\{i, j\}$

Walk of length L is a *sequence* $(i_0, i_1, \ldots, i_L)$ where each one of the vertices i_k is adjacent to i_{k+1} for all $k = 0, 1, \ldots, L-1$

Path of length L is a *non self-intersecting walk* passing through L edges of a graph.

Cycle (loop) of length L is a *closed path* along L edges of a graph which visits each vertex at most once.

Definitions



Simple, Undirected Graph G(N,M)
has N vertices i and M edges $\{i, j\}$

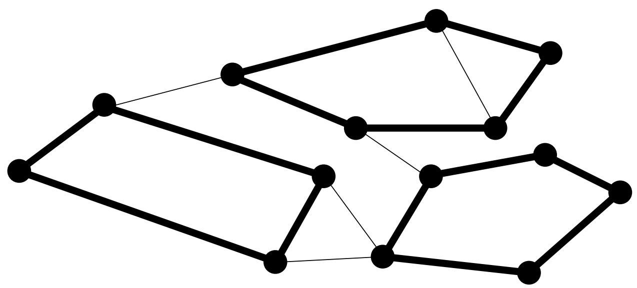
Walk of length L is a *sequence* $(i_0, i_1, \ldots, i_L)$ where each one of the vertices i_k is adjacent to i_{k+1} for all $k = 0, 1, \ldots, L-1$

Path of length L is a *non self-intersecting walk* passing through L edges of a graph.

Cycle (loop) of length L is a *closed path* along L edges of a graph which visits each vertex at most once.

Hamiltonian cycle $=$ cycle covering all vertices of a graph

Definitions



Simple, Undirected Graph G(N,M)
has N vertices i and M edges $\{i, j\}$

Walk of length L is a *sequence* $(i_0, i_1, \ldots, i_L)$ where each one of the vertices i_k is adjacent to i_{k+1} for all $k = 0, 1, \ldots, L-1$

Path of length L is a *non self-intersecting walk* passing through L edges of a graph.

Cycle (loop) of length L is a *closed path* along L edges of a graph which visits each vertex at most once.

Hamiltonian cycle = cycle covering all vertices of a graph

cycle cover = union of vertex disjoint cycles covering all vertices of a graph

Interest?

- Graph theory:

  Hamiltonian cycles (= cycles of length N): NP-complete (*cfr.* Traveling Salesman Problem)

  Statistical properties of # cycles on random graph ensembles

Interest?

- Graph theory:

  Hamiltonian cycles (= cycles of length N): NP-complete (*cfr.* Traveling Salesman Problem)

  Statistical properties of # cycles on random graph ensembles

- Understanding Real World Networks (e.g. Internet, WWW, biological networks, social networks):
  - local properties: degree distribution, clustering $\rightarrow$ short cycles
  - global properties: shortest paths, network motives $\rightarrow$ longer cycles
  - dynamics: feedback mechanism
  - vertex ranking

Computational Difficulty

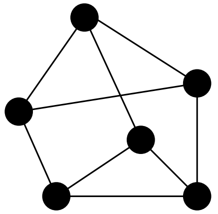
⇒ 3 fundamental questions:

1. Do they exist?
2. If yes, how many?
3. Can we locate them?

Computational Difficulty depends on length L of cycle:

- short cycles ($L = 3, 4, 5$): exhaustive enumeration has time upper bound of $\mathcal{O}(N \times \#\text{cycles})$, where $\#\text{cycles} \propto \exp N$
- intermediate cycles ($\lim_{N\to\infty} \frac{L}{N} = 0$): in limit $N \to \infty$ distribution can be computed for most random graph ensembles
- long extensive cycles ($L \propto N$), e.g., Hamiltonian cycles:

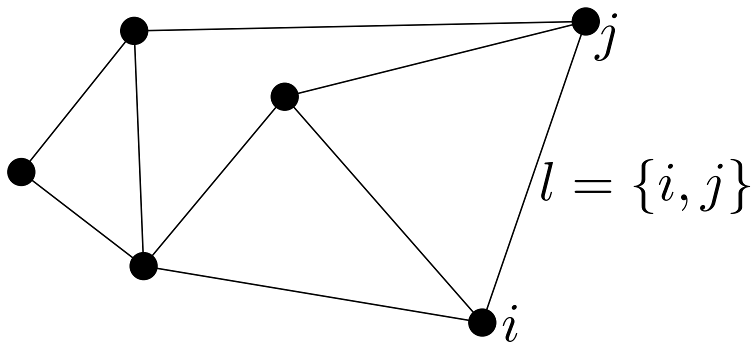
- Regular graphs: Hamiltonian with high probability (Wormald)
- Sparse graphs with minimum degree 3 and bounded maximum degree: conjectured to be Hamiltonian (Wormald)

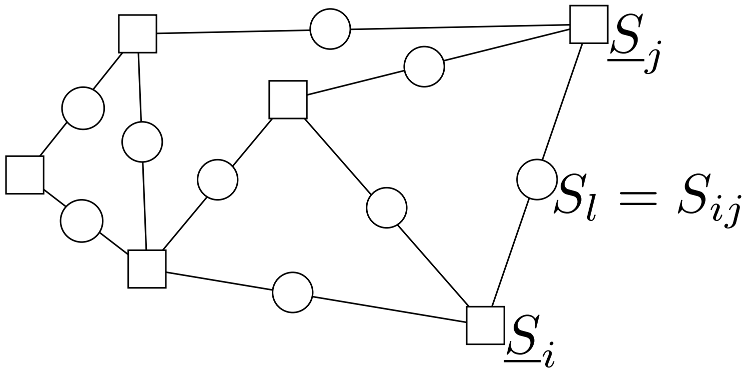


A Constraint Satisfaction Problem for Cycles

- $\forall$ edges l: $S_l = 0/1$ if edge l is absent / present

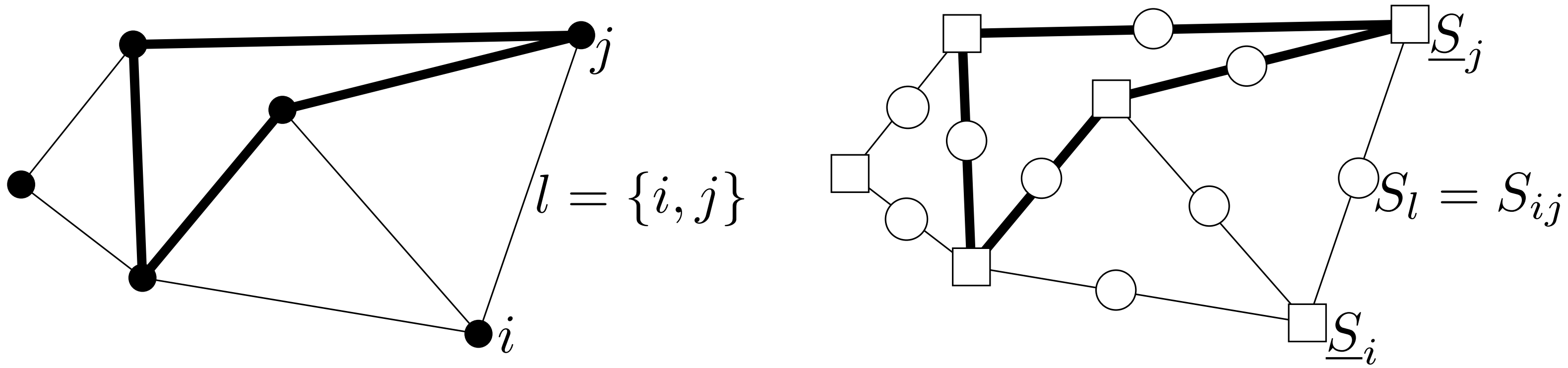
  $\forall$ vertices i: $\underline{S}_i = \{S_l | l$ is a neighboring edge of vertex $i\}$





A Constraint Satisfaction Problem for Cycles

- $\forall$ edges l: $S_l = 0/1$ if edge l is absent / present
  $\forall$ vertices i: $\underline{S}_i = \{S_l | l$ is a neighboring edge of vertex $i\}$

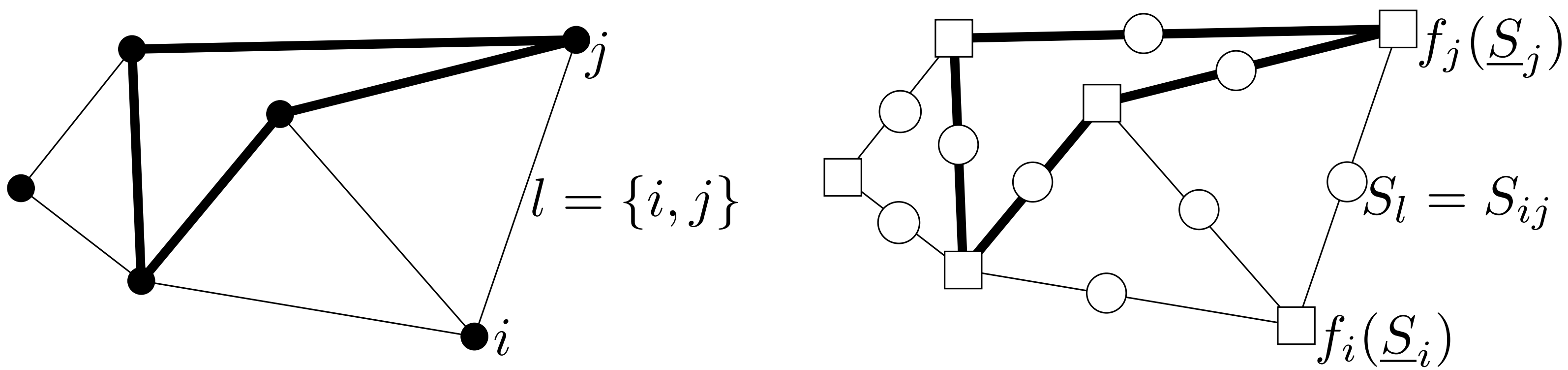


- Define $\mathrm{Prob}[\underline{S}] = \begin{cases} 0 & \text{if } \underline{S} \text{ is not a cycle} \\ f(u) & \text{if } \underline{S} \text{ is a cycle} \end{cases}$

A Constraint Satisfaction Problem for Cycles

- $\forall$ edges l: $S_l = 0/1$ if edge l is absent / present

  $\forall$ vertices i: $\underline{S}_i = \{S_l | l$ is a neighboring edge of vertex $i\}$



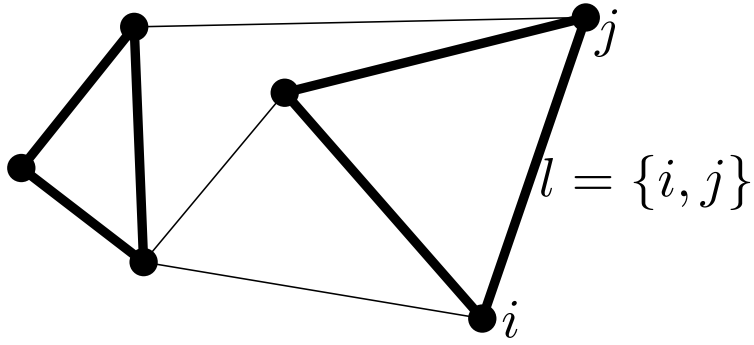
- Define $\text{Prob}[\underline{S}] = \begin{cases} 0 & \text{if } \underline{S} \text{ is not a cycle} \\ f(u) & \text{if } \underline{S} \text{ is a cycle} \end{cases}$

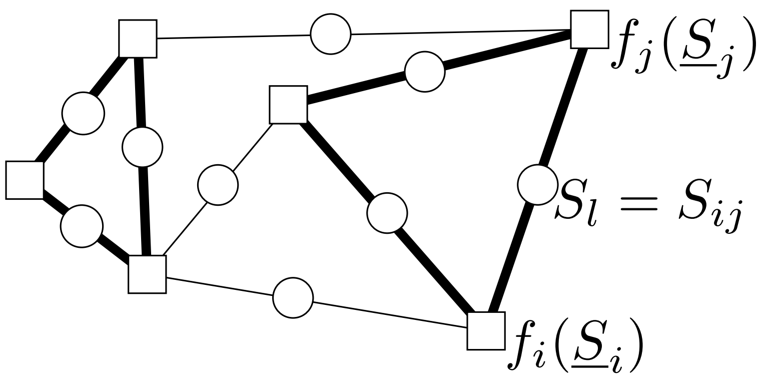
$$\text{Prob}[\underline{S}] = \frac{1}{Z} u^{\sum_l S_l} \prod_i f_i(\underline{S}_i) \quad \text{where} \quad f_i(\underline{S}_i) \begin{cases} 1 & \text{if } \sum_{l \in \partial i} S_l \in \{0, 2\} \\ 0 & \text{otherwise} \end{cases}$$

A Constraint Satisfaction Problem for Cycles

- $\forall$ edges l: $S_l = 0/1$ if edge l is absent / present

  $\forall$ vertices i: $\underline{S}_i = \{S_l | l$ is a neighboring edge of vertex $i\}$





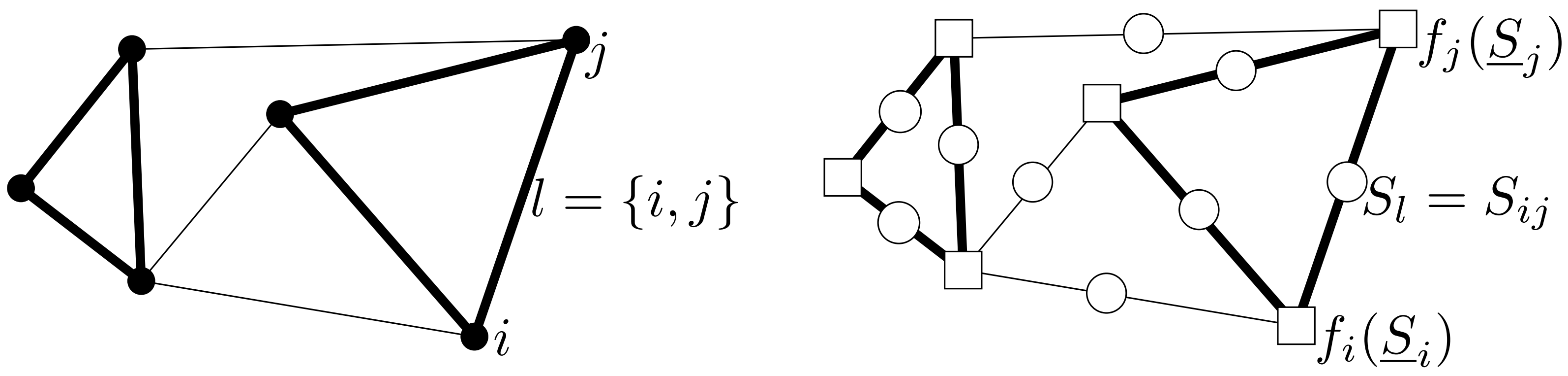
- Define $\mathrm{Prob}[\underline{S}] = \begin{cases} 0 & \text{if } \underline{S} \text{ is not a cycle} \\ f(u) & \text{if } \underline{S} \text{ is a cycle} \end{cases}$

$$\mathrm{Prob}[\underline{S}] = \frac{1}{Z} u^{\sum_l S_l} \prod_i f_i(\underline{S}_i) \quad \text{where} \quad f_i(\underline{S}_i) \begin{cases} 1 & \text{if } \sum_{l \in \partial i} S_l \in \{0,2\} \\ 0 & \text{otherwise} \end{cases}$$

A Constraint Satisfaction Problem for Cycles

- $\forall$ edges l: $S_l = 0/1$ if edge l is absent / present

  $\forall$ vertices i: $\underline{S}_i = \{S_l | l$ is a neighboring edge of vertex $i\}$



- Define $\text{Prob}[\underline{S}] = \begin{cases} 0 & \text{if } \underline{S} \text{ is not a cycle} \\ f(u) & \text{if } \underline{S} \text{ is a cycle} \end{cases}$

$$\text{Prob}[\underline{S}] = \frac{1}{Z} u^{\sum_l S_l} \prod_i f_i(\underline{S}_i) \quad \text{where} \quad f_i(\underline{S}_i) \begin{cases} 1 & \text{if } \sum_{l \in \partial i} S_l \in \{0, 2\} \\ 0 & \text{otherwise} \end{cases}$$

$u = 1$ uniform sampling

$u \to \infty$ cycles of longest length (e.g. Hamiltonian cycles)

I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

for $n = 1$ **to** M

- choose l_n: S_{l_n} is undefined
- draw S_{l_n} according to $\mathrm{Prob}[S_{l_n}|S_{l_1},\ldots,S_{l_{n-1}}]$

I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

for $n = 1$ **to** M

- choose l_n: S_{l_n} is undefined
- draw S_{l_n} according to $\mathrm{Prob}[S_{l_n}|S_{l_1}, \ldots, S_{l_{n-1}}]$

Problem 1: $\mathrm{Prob}[S_{l_n}|S_{l_1}, \ldots, S_{l_{n-1}}]$

Problem 2: probability law selecting set of cycles of total length L

I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

for $n = 1$ **to** M

- choose l_n: S_{l_n} is undefined
- draw S_{l_n} according to $\mathrm{Prob}[S_{l_n}|S_{l_1},\ldots,S_{l_{n-1}}]$

Problem 1: $\mathrm{Prob}[S_{l_n}|S_{l_1},\ldots,S_{l_{n-1}}]$

Problem 2: probability law selecting set of cycles of total length L

$$\mathrm{Prob}[\underline{S}] = \frac{1}{Z} u^{\sum_l S_l} \prod_i f_i(\underline{S}_i) \text{ where } f_i(\underline{S}_i) \begin{cases} 1 & \text{if } \sum_{l\in\partial i} S_l \in \{0,2\} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{for } u \to \infty \Rightarrow \begin{cases} \text{cycle cover} & \text{if } \underline{S} \text{ consists of more than one cycle} \\ \text{hamiltonian cycle} & \text{if } \underline{S} \text{ consists of just one cycle} \end{cases}$$

I.1 Decimation ⇒ Hamiltonian Cycles

for $n = 1$ **to** M

- choose l_n: S_{l_n} is undefined
- draw S_{l_n} according to $\mathrm{Prob}[S_{l_n}|S_{l_1}, \dots, S_{l_{n-1}}]$

Problem 1: $\mathrm{Prob}[S_{l_n}|S_{l_1}, \dots, S_{l_{n-1}}]$

$\rightarrow$ approximate by means of Belief Propagation $\Leftrightarrow \mathrm{Prob}[\underline{S}] = \prod g(\underline{S}_x)$

Problem 2: probability law selecting set of cycles of total length L

$$\mathrm{Prob}[\underline{S}] = \frac{1}{Z} u^{\sum_l S_l} \prod_i f_i(\underline{S}_i) \text{ where } f_i(\underline{S}_i) \begin{cases} 1 & \text{if } \sum_{l \in \partial i} S_l \in \{0, 2\} \\ 0 & \text{otherwise} \end{cases}$$

$$\text{for } u \rightarrow \infty \Rightarrow \begin{cases} \text{cycle cover} & \text{if } \underline{S} \text{ consists of more than one cycle} \\ \text{hamiltonian cycle} & \text{if } \underline{S} \text{ consists of just one cycle} \end{cases}$$

Belief Propagation

Compute partition function $Z = \sum_{\underline{x}} w(\underline{x})$

$\Leftrightarrow$ Minimizing the corresponding Gibbs free energy functional

$$F_{\text{Gibbs}}[p_{\text{var}}] = \sum_{\underline{x}} p_{\text{var}}(\underline{x}) \ln \left(\frac{p_{\text{var}}(\underline{x})}{w(\underline{x})} \right)$$

since $\min_{p_{\text{var}}} F_{\text{Gibbs}}[p_{\text{var}}] = F_{\text{Gibbs}}[P_{\text{Gibbs}}] = -\ln Z.$

Mean Field approximation: factorizable trial distributions

$p_{\text{MF}}(\underline{x}) = \prod_i p_i(x_i)$

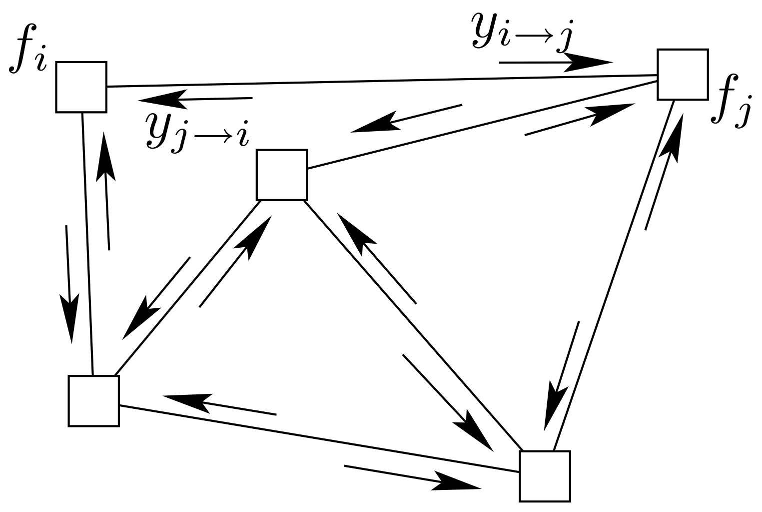
Bethe approximation: take first order correlations into account

e.g. $p_{\text{Bethe}}(\underline{x}) = \frac{\prod_{\{i,j\}} p_{ij}(x_i, x_j)}{\prod_i p_i(x_i)}$ demanding normalized distributions p_i, p_{ij} and consistency

$\Rightarrow$ Introduce Lagrange Multipliers

$\Leftrightarrow$ Finding fixed point of the corresponding distributed Belief Propagation (BP) algorithm.

Belief Propagation



- Initialize messages $y_{i\to j}$ randomly.
- Iterate BP until convergence, where each update takes up a time $\mathcal{O}(M)$:

$$y_{i\to j} = f_1\left(u, \{y_{k\to i}\}_{k\in\partial i\setminus j}\right)$$

$$\Rightarrow \quad p_l(S_l = 1) = \frac{u y_{i\to j} y_{j\to i}}{1+u y_{i\to j} y_{j\to i}}$$

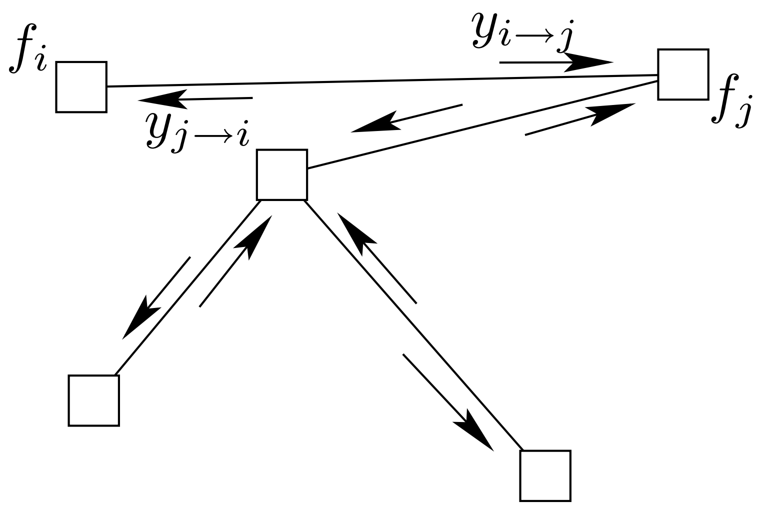
On a tree-like graph:

- BP converges fast!
- F_{Bethe}, and thus BP, is exact!

On a general graph with cycles:

- In theory, BP does not necessarily converge, but in practice it often does after a reasonable amount of iterations.
  $\Rightarrow$ Allows to investigate larger graphs $\sim \mathcal{O}(10^6)$.

Belief Propagation



- Initialize messages $y_{i\to j}$ randomly.
- Iterate BP until convergence, where each update takes up a time $\mathcal{O}(M)$:

$$y_{i\to j} = f_1\left(u, \{y_{k\to i}\}_{k\in\partial i\setminus j}\right)$$

$$\Rightarrow \quad p_l(S_l = 1) = \frac{u y_{i\to j} y_{j\to i}}{1+u y_{i\to j} y_{j\to i}}$$

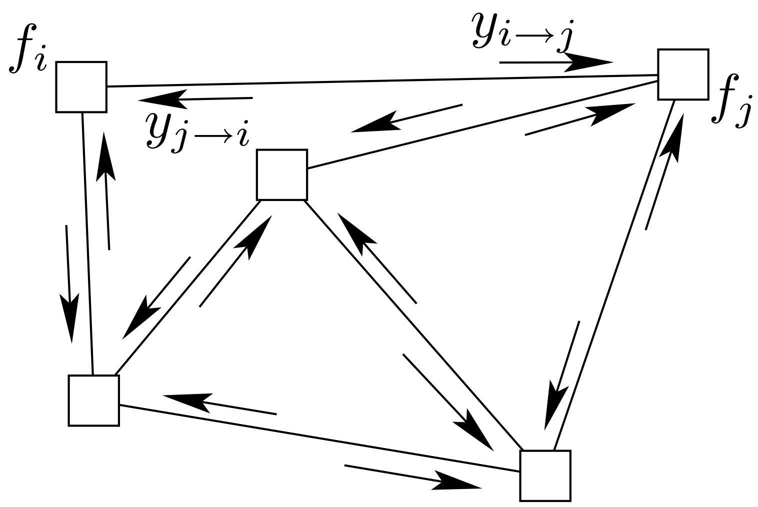
On a tree-like graph:

- BP converges fast!
- F_{Bethe}, and thus BP, is exact!

On a general graph with cycles:

- In theory, BP does not necessarily converge, but in practice it often does after a reasonable amount of iterations.
  $\Rightarrow$ Allows to investigate larger graphs $\sim \mathcal{O}(10^6)$.

Belief Propagation



- Initialize messages $y_{i\to j}$ randomly.
- Iterate BP until convergence, where each update takes up a time $\mathcal{O}(M)$:

$$y_{i\to j} = f_1\left(u, \{y_{k\to i}\}_{k\in\partial i\setminus j}\right)$$

$$\Rightarrow \quad p_l(S_l = 1) = \frac{u y_{i\to j} y_{j\to i}}{1 + u y_{i\to j} y_{j\to i}}$$

On a tree-like graph:

- BP converges fast!
- F_{Bethe}, and thus BP, is exact!

On a general graph with cycles:

- In theory, BP does not necessarily converge, but in practice it often does after a reasonable amount of iterations.
  $\Rightarrow$ Allows to investigate larger graphs $\sim \mathcal{O}(10^6)$.

I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

- Performance on sparse graphs with $N = 100, 200, \ldots, 1600$
  - Regular graphs ($c = 3, 4, 5$): $\forall$ $\mathbf{HC}$
  - Bimodal graphs ($q_{3,4}^{0.5}$,$q_{3,5}^{0.5}$,$q_{4,5}^{0.5}$): $94 - 99\%$ $\mathbf{HC}$ ($\pm 99\%$ $\mathbf{CC}$)

	$q_{3,4}^{0.5}$		$q_{3,5}^{0.5}$		$q_{4,5}^{0.5}$	
	CC	HC	CC	HC	CC	HC
N	DEC		DEC		DEC	
100	99.9	96.0	98.9	69.9	98.7	56.9
200	99.6	96.2	99.7	71.1	98.9	50.0
400	99.7	96.4	99.9	67.7	98.9	50.7
800	99.8	96.7	99.6	68.9	99.6	46.8
1600	99.7	97.8	99.9	68.6	99.9	52.3

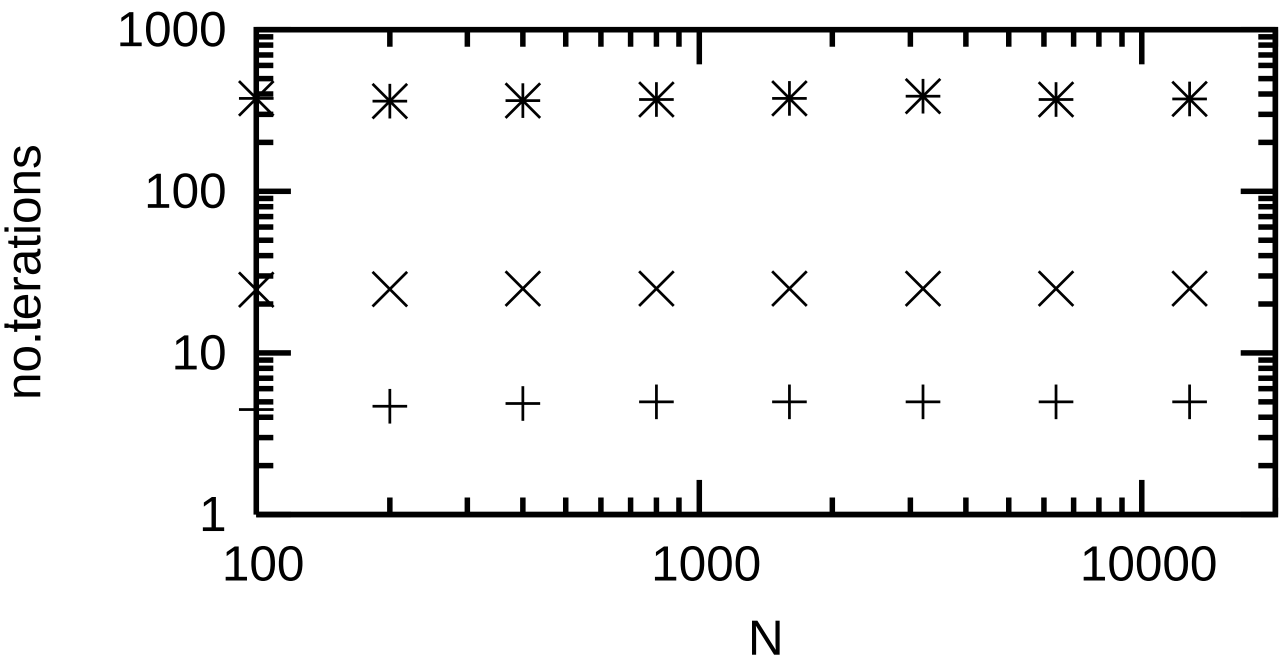
I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

- Performance on sparse graphs with $N = 100, 200, \ldots, 1600$
  - Regular graphs ($c = 3, 4, 5$): $\forall$ $\mathbf{HC}$
  - Bimodal graphs ($q^{0.5}_{3,4}$,$q^{0.5}_{3,5}$,$q^{0.5}_{4,5}$): $94 - 99\%$ $\mathbf{HC}$ ($\pm 99\%$ $\mathbf{CC}$)

N	$q^{0.5}_{3,4}$ CC	$q^{0.5}_{3,4}$ HC DEC	$q^{0.5}_{3,4}$ HC LR	$q^{0.5}_{3,5}$ CC	$q^{0.5}_{3,5}$ HC DEC	$q^{0.5}_{3,5}$ HC LR	$q^{0.5}_{4,5}$ CC	$q^{0.5}_{4,5}$ HC DEC	$q^{0.5}_{4,5}$ HC LR
100	99.9	96.0	99.6	98.9	69.9	92.9	98.7	56.9	96.0
200	99.6	96.2	99.3	99.7	71.1	95.2	98.9	50.0	96.0
400	99.7	96.4	99.2	99.9	67.7	95.4	98.9	50.7	94.2
800	99.8	96.7	98.7	99.6	68.9	95.7	99.6	46.8	94.5
1600	99.7	97.8	98.7	99.9	68.6	92.0	99.9	52.3	94.0

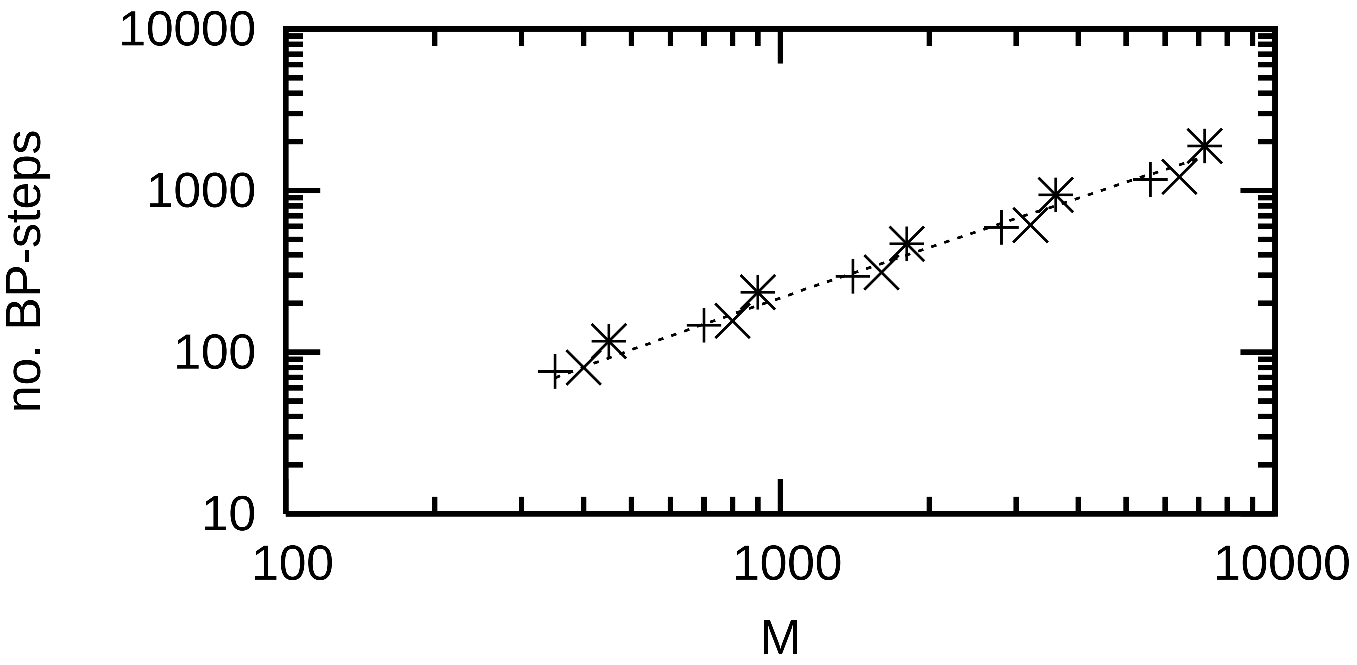
I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

- Time complexity
  - decimation procedure $\sim \mathcal{O}(M^2)$
    e.g. $q_c(k) = \delta_{k,c}$: $c = 3(+), 4(\times), 5(*)$



I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

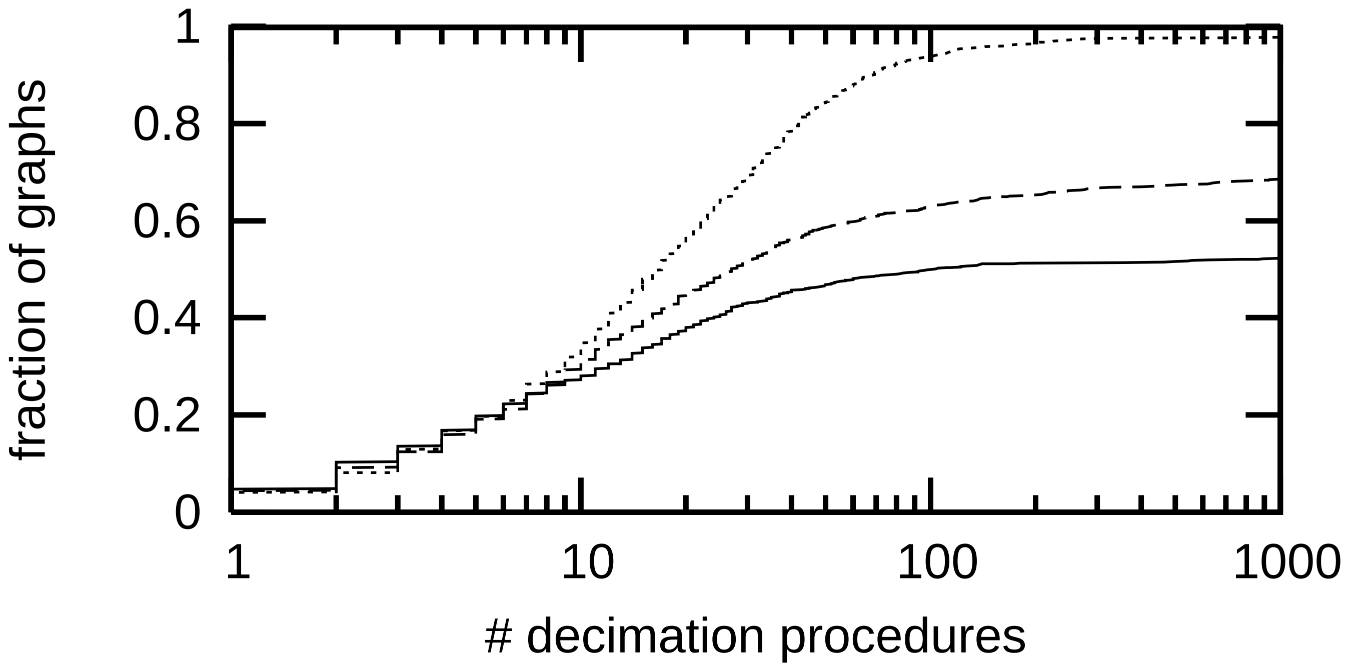
- Time complexity
  - decimation procedure $\sim \mathcal{O}(M^2)$
    e.g. $q_{3,4}^{0.5}(+), q_{3,5}^{0.5}(\times), q_{4,5}^{0.5}(*)$



slope $\simeq 0.23$

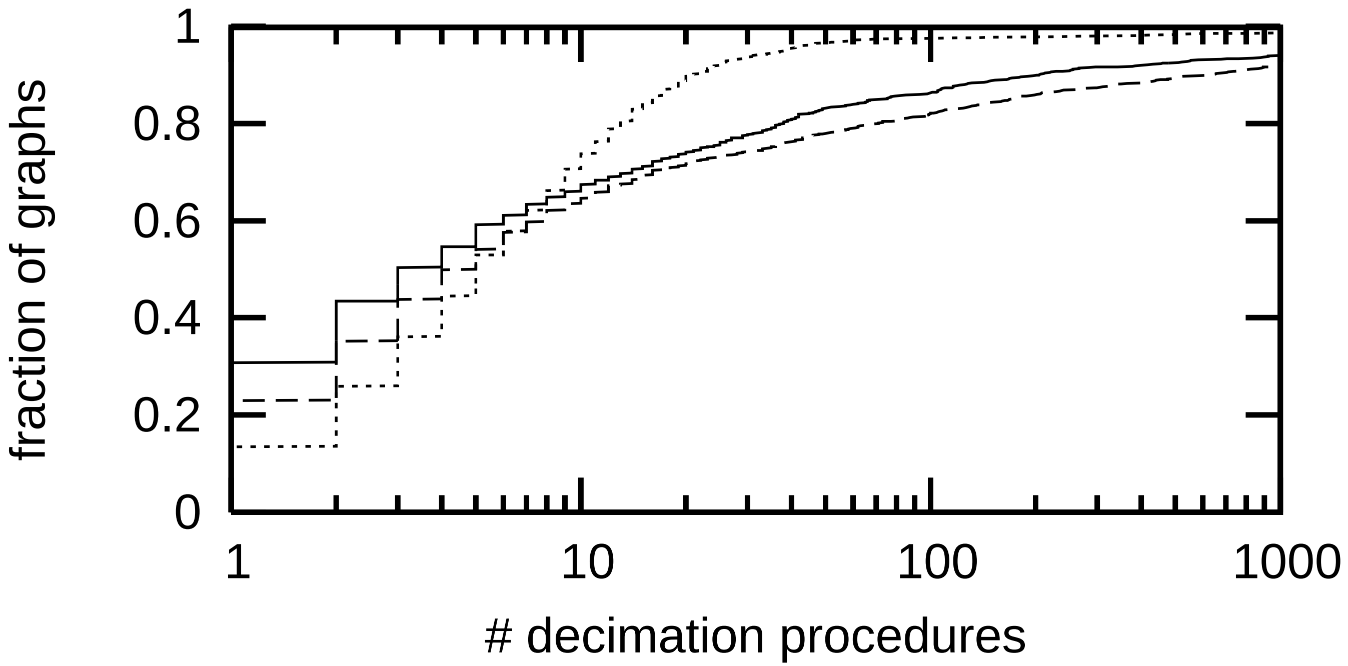
I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

- Time complexity
  - decimation procedure $\sim \mathcal{O}(M^2)$
  - number of trials
    e.g. $q_{3,4}^{0.5}$ (dotted curve), $q_{3,5}^{0.5}$ (dashed curve), $q_{4,5}^{0.5}$ (full line)



I.1 Decimation $\Rightarrow$ Hamiltonian Cycles

- Time complexity
  - decimation procedure $\sim \mathcal{O}(M^2)$
  - number of trials
    e.g. $q_{3,4}^{0.5}$ (dotted curve), $q_{3,5}^{0.5}$ (dashed curve), $q_{4,5}^{0.5}$ (full line)



Optimization: Local rewiring $\Rightarrow$ CC $\rightarrow$ HC

I.2 Markov Chain Monte Carlo Sampling

Ergodic, fast mixing Markov Chain $\underline{S}, \underline{S}', \underline{S}'', \ldots$, which admits $\mathrm{Prob}[\underline{S}]$ as unique stationary distribution.

$\rightarrow$ Ergodic? Convergence time?

$\rightarrow$ Determine appropriate transitions $\underline{S} \rightarrow \underline{S}'$, and transition rates $W(\underline{S} \rightarrow \underline{S}')$: e.g. by means of detailed balance:

$$W(\underline{S} \rightarrow \underline{S}')\mathrm{Prob}[\underline{S}] = W(\underline{S}' \rightarrow \underline{S})\mathrm{Prob}[\underline{S}']$$

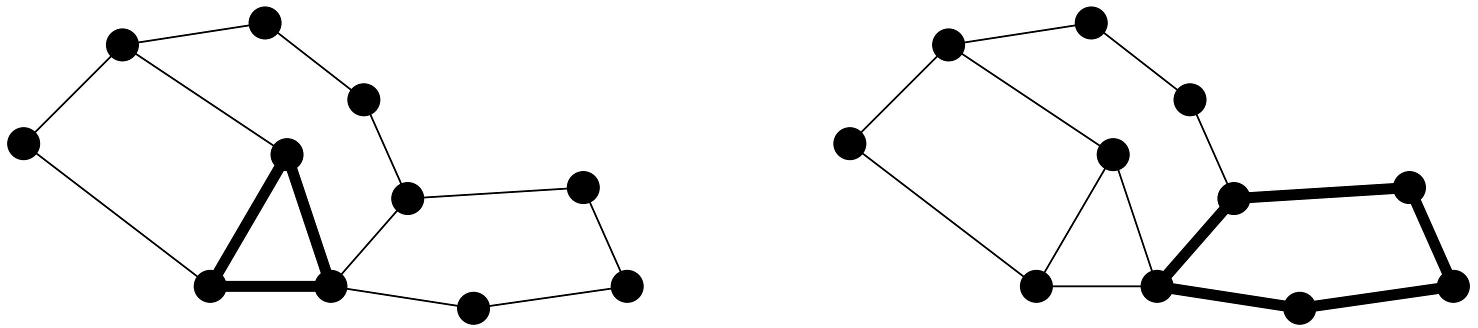
I.2 Markov Chain Monte Carlo Sampling

Ergodic, fast mixing Markov Chain $\underline{S}, \underline{S}', \underline{S}'', \ldots$, which admits $\mathrm{Prob}[\underline{S}]$ as unique stationary distribution.

$\rightarrow$ Ergodic? Convergence time?

$\rightarrow$ Determine appropriate transitions $\underline{S} \rightarrow \underline{S}'$, and transition rates $W(\underline{S} \rightarrow \underline{S}')$: e.g. by means of detailed balance:
$W(\underline{S} \rightarrow \underline{S}')\mathrm{Prob}[\underline{S}] = W(\underline{S}' \rightarrow \underline{S})\mathrm{Prob}[\underline{S}']$

$$\mathrm{Prob}[\underline{S}] = \begin{cases} 0 & \text{if } \underline{S} \text{ is not a cycle} \\ f(u) & \text{if } \underline{S} \text{ is a cycle} \end{cases} = \frac{1}{Z} u^{\sum_l S_l} \prod_i f_i(\underline{S}_i)$$



I.2 Markov Chain Monte Carlo Sampling

Ergodic, fast mixing Markov Chain $\underline{S}, \underline{S}', \underline{S}'', \ldots$, which admits $\text{Prob}[\underline{S}]$ as unique stationary distribution.

→ Ergodic? Convergence time?

→ Determine appropriate transitions $\underline{S} \to \underline{S}'$, and transition rates $W(\underline{S} \to \underline{S}')$: e.g. by means of detailed balance:
$W(\underline{S} \to \underline{S}')\text{Prob}[\underline{S}] = W(\underline{S}' \to \underline{S})\text{Prob}[\underline{S}']$

$$\text{Prob}[\underline{S}] = \begin{cases} 0 & \text{if } \underline{S} \text{ is not a path} \\ f(u) & \text{if } \underline{S} \text{ is a path} \end{cases} = \frac{1}{Z}\left(u^{\sum_l S_l}\right)\left(\prod_i \tilde{f}_i(\underline{S}_i)\right)\left(\eta^{n_{\underline{S}}}\right)$$

$n_{\underline{S}} =$ number of disjoint paths of configuration S

$\eta \in [0,1)$

$$\tilde{f}_i(\underline{S}_i) = \begin{cases} 1 & \text{if } \sum_{l\in\partial i} S_l \in \{0,2\} \\ \epsilon \in [0,1] & \text{if } \sum_{l\in\partial i} S_l = 1 \\ 0 & \text{otherwise} \end{cases}$$

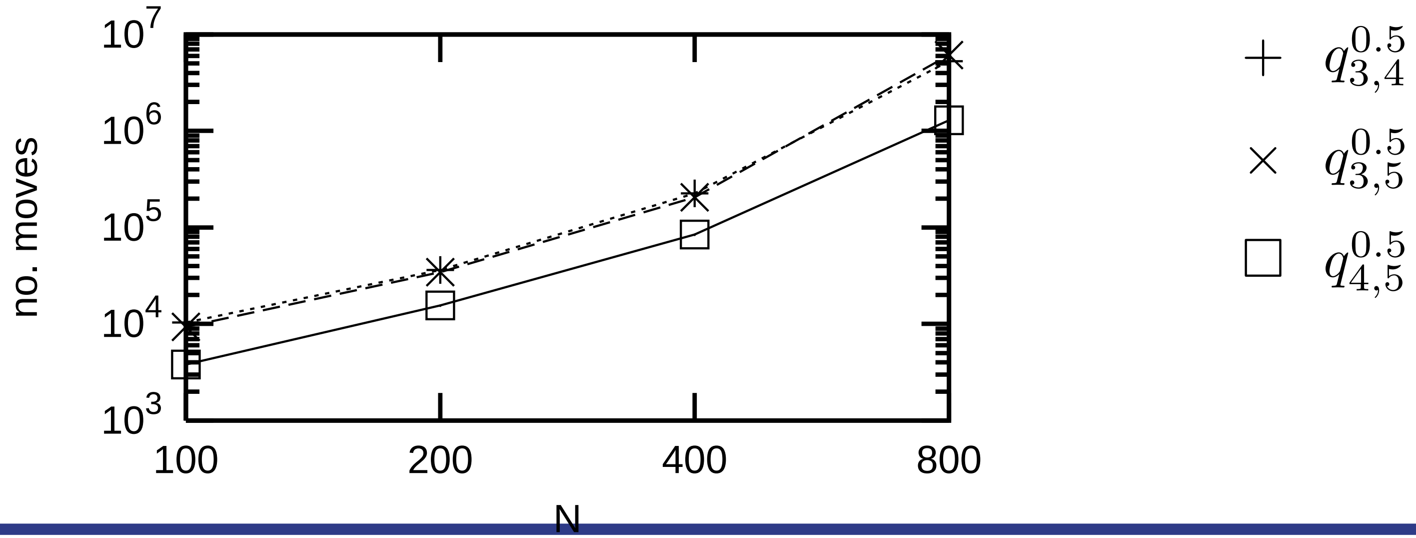
I.2 Monte Carlo $\Rightarrow$ Hamiltonian Cycles

- Succes rate:
  - Regular graphs of size $N = 100, 200, 400, 800 : 100\%$
  - Bimodal graphs ($q^{0.5}_{3,4}$,$q^{0.5}_{3,5}$,$q^{0.5}_{4,5}$) of size $N = 100, 200, 400, 800 : 100\% \rightarrow$ Comfirmation of Wormald's conjecture on non-regular graphs

I.2 Monte Carlo $\Rightarrow$ Hamiltonian Cycles

- Succes rate:
  - Regular graphs of size $N = 100, 200, 400, 800 : 100\%$
  - Bimodal graphs ($q^{0.5}_{3,4}$,$q^{0.5}_{3,5}$,$q^{0.5}_{4,5}$) of size $N = 100, 200, 400, 800 : 100\% \rightarrow$ Comfirmation of Wormald's conjecture on non-regular graphs

- Time requirements $\rightarrow$ optimized by means of N-fold MC (up to M times faster):
  - Distribution depends on u, ϵ and η
  -



Comparison

We find *Hamiltonian Cycles* for all sparse graphs with $k_{\min} = 3$.

	BP		**MC**
+	versatile	-	very parameter sensitive
+	polynomial in N	-	exponential in N
-	no garantee	+	more reliable

Comparison

We find *Hamiltonian Cycles* for all sparse graphs with $k_{\min} = 3$.

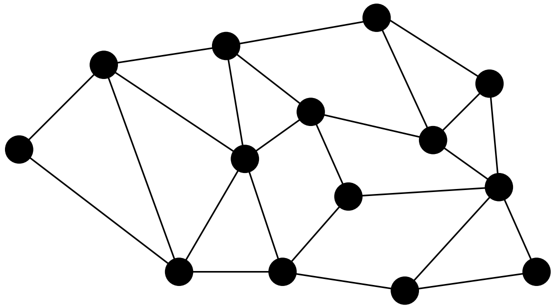
BP		MC	
+	versatile	-	very parameter sensitive
+	polynomial in N	-	exponential in N
-	no garantee	+	more reliable

$\rightarrow$ CPU time: e.g. bimodal graph with $q_{3,4}^{0.5}$, $N = 1600$

BP 30’, i.e. 72 trials (70 cycle covers) (with local moves: 5’)

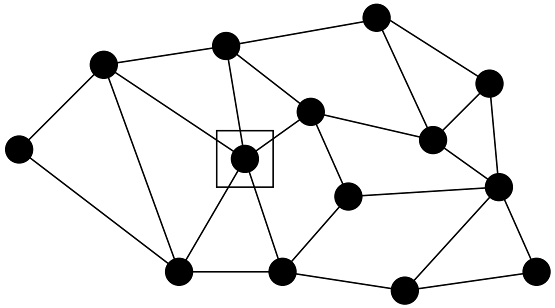
MC 40’ (with optmized parameter values)

II. Vertex (and Edge) Ranking



Ranking is an *objective (topology based) measure of importance* of the vertices of a graph

II. Vertex (and Edge) Ranking

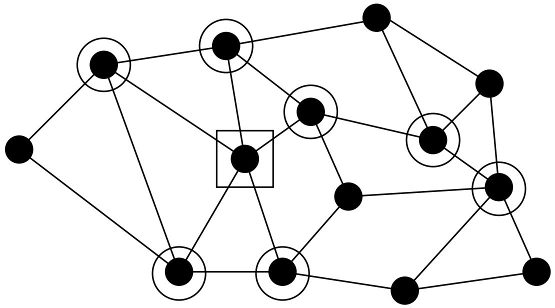


Ranking is an *objective (topology based) measure of importance* of the vertices of a graph

Degree $\mathcal{D}(i) = |\partial i|$

(+) easy to compute (-) very rough measure

II. Vertex (and Edge) Ranking

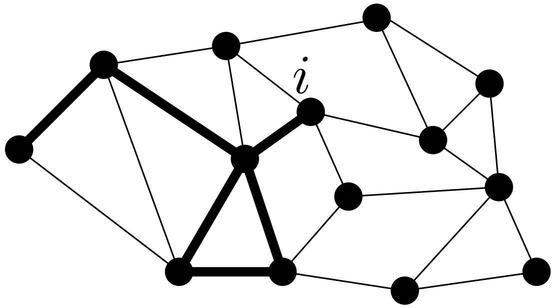


Ranking is an *objective (topology based) measure of importance* of the vertices of a graph

Degree $\mathcal{D}(i) = |\partial i|$

(+) easy to compute (-) very rough measure

II. Vertex (and Edge) Ranking



Ranking is an *objective (topology based) measure of importance* of the vertices of a graph

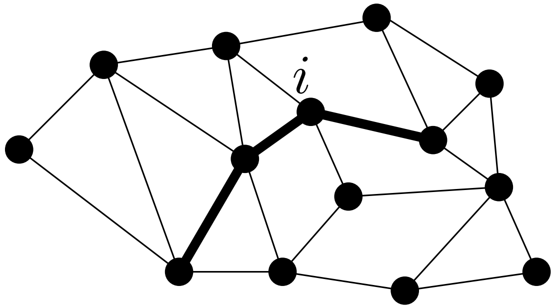
Degree $\mathcal{D}(i) = |\partial i|$

(+) easy to compute (-) very rough measure

PageRank $\mathcal{P}(i) \propto d \sum_{j \in \partial_i^+} \frac{\mathcal{P}(j)}{d_j^-}$

(+) iterative algorithm, emulates behavior of a Random Walk

II. Vertex (and Edge) Ranking



Ranking is an *objective (topology based) measure of importance* of the vertices of a graph

Degree $\mathcal{D}(i) = |\partial i|$

(+) easy to compute (-) very rough measure

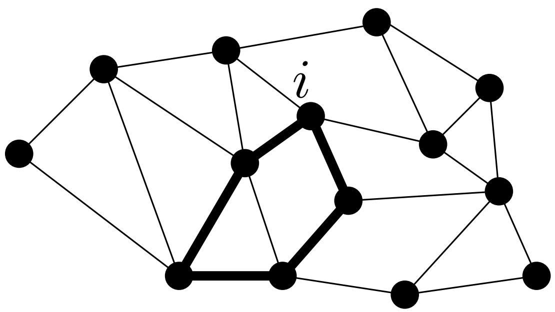
PageRank $\mathcal{P}(i) \propto d \sum_{j \in \partial_i^+} \frac{\mathcal{P}(j)}{d_j^-}$

(+) iterative algorithm, emulates behavior of a Random Walk

Betweenness Centrality $\mathcal{B}(i) = \sum_{k,l(\neq i) \in V} \frac{\sigma_{k,l}(i)}{\sigma_{k,l}}$

(+) based on shortest paths, (-) time requirements $\sim \mathcal{O}(NM)$

II. Vertex (and Edge) Ranking



Ranking is an *objective (topology based) measure of importance* of the vertices of a graph

Degree $\mathcal{D}(i) = |\partial i|$

(+) easy to compute (-) very rough measure

PageRank $\mathcal{P}(i) \propto d \sum_{j \in \partial_i^+} \frac{\mathcal{P}(j)}{d_j^-}$

(+) iterative algorithm, emulates behavior of a Random Walk

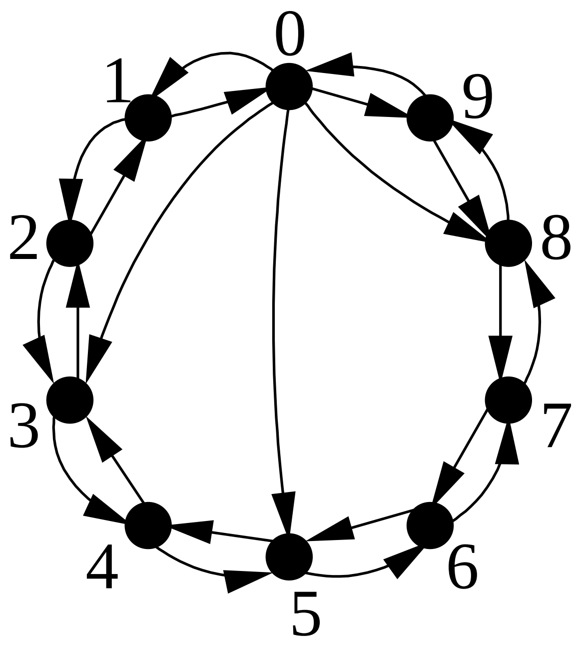
Betweenness Centrality $\mathcal{B}(i) = \sum_{k,l(\neq i) \in V} \frac{\sigma_{k,l}(i)}{\sigma_{k,l}}$

(+) based on shortest paths, (-) time requirements $\sim \mathcal{O}(NM)$

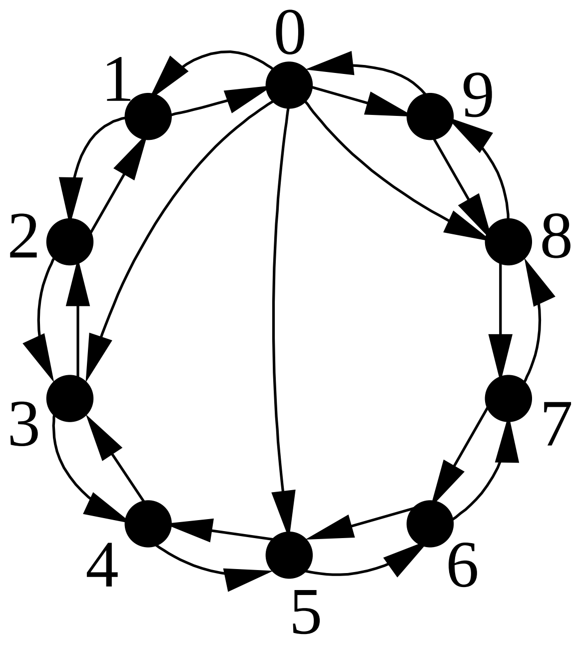
Loop Ranking $\mathcal{L}(i) = \sum_{i \in \mathrm{Cycle}} w(\mathrm{Cycle}) \propto \mathrm{Prob}(i \in \mathrm{Cycle})$

for $\mathrm{Prob}[\underline{S}] = \frac{1}{Z} \prod_l (r_l)^{S_l} \prod_i f_i(\underline{S}_i)$

Directed Small World Network



Directed Small World Network

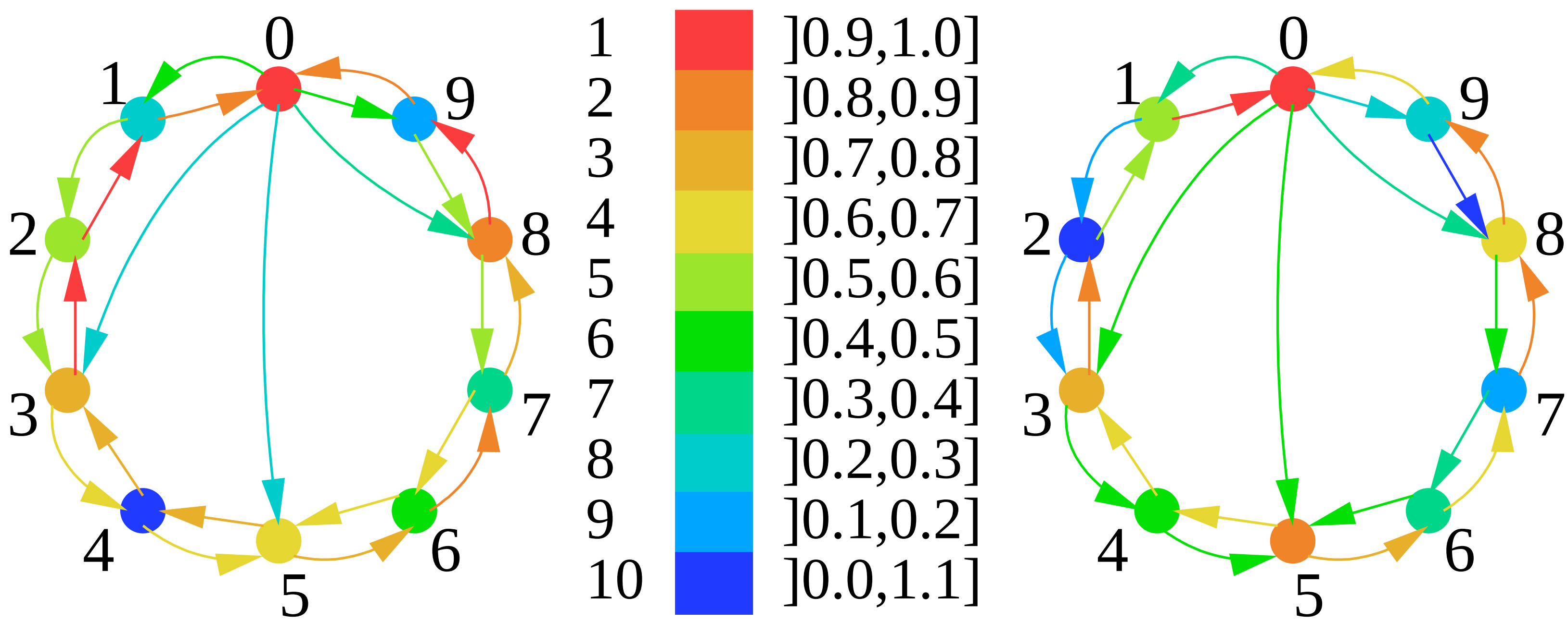


R	i	$\frac{\mathcal{P}(i)}{\mathcal{P}(5)}$	i	$\frac{\mathcal{L}(i)}{\mathcal{L}(0)}$	i	$\frac{\mathcal{B}(i)}{\mathcal{B}(0)}$
1	5	1.00	0	1.00	0	1.00
2	4	0.94	8	0.92	5	0.68
3	3	0.91	3	0.90	3	0.58
4	6	0.90	5	0.89	8	0.55
5	7	0.84	2	0.87	1	0.50
6	8	0.78	6	0.84	4	0.45
7	2	0.71	7	0.83	6	0.45
8	0	0.53	1	0.82	9	0.43
9	9	0.53	9	0.82	7	0.42
10	1	0.49	4	0.81	2	0.39

Directed Small World Network

Loop Ranking

Betweenness Centrality



Path-based Ranking:

- capture importance of vertices on small-world networks
- allow for edge ranking
- lead to similar results for the most important vertices and edges

Conclusions and Future Perspectives

- We find Hamiltonian cycles on regular and non-regular sparse graphs,
  - b.m.o. BP: faster
  - b.m.o. MC: more reliable
- New path-based vertex and edge ranking captures their importance in traffic flow (on directed small world networks).

Conclusions and Future Perspectives

- We find Hamiltonian cycles on regular and non-regular sparse graphs,
  - b.m.o. BP: faster
  - b.m.o. MC: more reliable
- New path-based vertex and edge ranking captures their importance in traffic flow (on directed small world networks).

$\rightarrow$ Deeper investigation of the level of approximation of BP.

$\rightarrow$ Improve MC by finding optimal parameters in automated way.

$\rightarrow$ Find loops or paths of intermediate length.

$\rightarrow$ Investigate real-world networks (scale free, weighted).

$\rightarrow$ Consider a Potts-like configuration space.