Cycles in Random Graphs

Valery Van Kerrebroeck

Enzo Marinari, Guilhem Semerjian

[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 
  \((cfr. \text{ Traveling Salesman Problem})\)
  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 $i$: $S_i = \{S_l | l$ is a neighboring edge of vertex $i\}$
A Constraint Satisfaction Problem for Cycles

- ∀ edges \( l \): \( S_l = 0/1 \) if edge \( l \) is absent / present
- ∀ vertices \( i \): \( S_i = \{ S_l \mid l \text{ is a neighboring edge of vertex } i \} \)

Define \( \text{Prob} [S] = \begin{cases} 0 & \text{if } S \text{ is not a cycle} \\ f(u) & \text{if } S \text{ is a cycle} \end{cases} \)
A Constraint Satisfaction Problem for Cycles

- ∀ edges \( l \): \( S_l = 0/1 \) if edge \( l \) is absent / present
- ∀ vertices \( i \): \( S_i = \{S_l | l \) is a neighboring edge of vertex \( i \}\)

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

\[
\text{Prob}[S] = \frac{1}{Z} u \sum_i S_i \prod_i f_i(S_i) \quad \text{where} \quad f_i(S_i) = \begin{cases} 1 & \text{if } \sum_{l \in \partial_i} S_l \in \{0, 2\} \\ 0 & \text{otherwise} \end{cases}
\]
A Constraint Satisfaction Problem for Cycles

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

- \forall \text{ vertices } i: S_i = \{S_l | l \text{ is a neighboring edge of vertex } i\}

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

\[
\text{Prob}[S] = \frac{1}{Z} e^{\sum_i S_i} \prod_i f_i(S_i) \quad \text{where } f_i(S_i) = \begin{cases} 1 & \text{if } \sum_{l \in \partial_i} S_l \in \{0, 2\} \\ 0 & \text{otherwise} \end{cases}
\]
A Constraint Satisfaction Problem for Cycles

- For all edges $l$: $S_l = 0/1$ if edge $l$ is absent / present
- For all vertices $i$: $S_i = \{S_l | l \text{ is a neighboring edge of vertex } i\}$

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

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

$u = 1$ 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 $\text{Prob}[S_{l_n} | S_{l_1}, \ldots, S_{l_{n-1}}]$
I.1 Decimation ⇒ Hamiltonian Cycles

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

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

**Problem 2:** probability law selecting set of cycles of total length $L$
I.1 Decimation ⇒ Hamiltonian Cycles

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

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

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

$$\text{Prob} [S] = \frac{1}{Z} u \sum_{l} S_l \prod_{i} f_i(S_i)$$
where $f_i(S_i) \begin{cases} 1 & \text{if } \sum_{l \in \partial_i} S_l \in \{0, 2\} \\ 0 & \text{otherwise} \end{cases}$

for $u \to \infty \Rightarrow \begin{cases} \text{cycle cover} & \text{if } S \text{ consists of more than one cycle} \\ \text{hamiltonian cycle} & \text{if } S \text{ consists of just one cycle} \end{cases}$
I.1 Decimation ⇒ Hamiltonian Cycles

for \( n = 1 \) to \( M \)
- choose \( l_n \): \( S_{l_n} \) is undefined
- draw \( S_{l_n} \) according to \( \text{Prob}[S_{l_n}|S_{l_1}, \ldots, S_{l_{n-1}}] \)

Problem 1: \( \text{Prob}[S_{l_n}|S_{l_1}, \ldots, S_{l_{n-1}}] \)

→ approximate by means of Belief Propagation ⇒ \( \text{Prob}[S] = \prod g(S_x) \)

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

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

for \( u \to \infty \) ⇒ \( \begin{cases} 
\text{cycle cover} & \text{if } S \text{ consists of more than one cycle} \\
\text{hamiltonian cycle} & \text{if } S \text{ consists of just one cycle}
\end{cases} \)
Belief Propagation

Compute partition function $Z = \sum x w(x)$

$\Leftrightarrow$ Minimizing the corresponding Gibbs free energy functional

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

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

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

Bethe approximation: take first order correlations into account

e.g. $p_{\text{Bethe}}(x) = \frac{\prod_{\{i,j\}} p_{ij}(x_i, x_j)}{\prod_i p_i(x_i)}$ 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 $y_{i\rightarrow j}$ randomly.
- Iterate BP until convergence, where each update takes up a time $O(M)$:
  
  $y_{i\rightarrow j} = f_1(u, \{y_{k\rightarrow i}\}_{k \in \partial i \setminus j})$

  $\Rightarrow p_i(S_i = 1) = \frac{uy_{i\rightarrow j}y_{j\rightarrow i}}{1 + uy_{i\rightarrow j}y_{j\rightarrow i}}$

On a tree-like graph:
- BP converges fast!
- $F_{\text{Bethe}}$, 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 $\sim O(10^6)$. 
Belief Propagation

- Initialize messages $y_{i \rightarrow j}$ randomly.
- Iterate BP until convergence, where each update takes up a time $O(M)$:
  $y_{i \rightarrow j} = f_1 \left( u, \{ y_{k \rightarrow i} \}_{k \in \partial i \setminus j} \right)$

  $p_I(S_I = 1) = \frac{uy_{i \rightarrow j} y_{j \rightarrow i}}{1 + uy_{i \rightarrow j} y_{j \rightarrow i}}$

On a tree-like graph:
- BP converges fast!
- $F_{\text{Bethe}}$, 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 $\sim O(10^6)$. 
Belief Propagation

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

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

$$\Rightarrow p_l(S_l = 1) = \frac{uy_{i \rightarrow j}y_{j \rightarrow i}}{1 + uy_{i \rightarrow j}y_{j \rightarrow i}}$$

On a tree-like graph:
- BP converges fast!
- $F_{\text{Bethe}}$, 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 $\sim O(10^6)$. 
I.1 Decimation ⇒ Hamiltonian Cycles

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

<table>
<thead>
<tr>
<th>N</th>
<th>$q_{3,4}^{0.5}$</th>
<th>$q_{3,5}^{0.5}$</th>
<th>$q_{4,5}^{0.5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>HC</td>
<td>DEC</td>
</tr>
<tr>
<td>100</td>
<td>99.9</td>
<td>96.0</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>99.6</td>
<td>96.2</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>99.7</td>
<td>96.4</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>99.8</td>
<td>96.7</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>99.7</td>
<td>97.8</td>
<td></td>
</tr>
</tbody>
</table>
I.1 Decimation ⇒ Hamiltonian Cycles

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

<table>
<thead>
<tr>
<th>N</th>
<th>( q_{3,4}^{0.5} )</th>
<th>( q_{3,5}^{0.5} )</th>
<th>( q_{4,5}^{0.5} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CC</td>
<td>HC</td>
<td>DEC</td>
</tr>
<tr>
<td>100</td>
<td>99.9</td>
<td>96.0</td>
<td>99.6</td>
</tr>
<tr>
<td>200</td>
<td>99.6</td>
<td>96.2</td>
<td>99.3</td>
</tr>
<tr>
<td>400</td>
<td>99.7</td>
<td>96.4</td>
<td>99.2</td>
</tr>
<tr>
<td>800</td>
<td>99.8</td>
<td>96.7</td>
<td>98.7</td>
</tr>
<tr>
<td>1600</td>
<td>99.7</td>
<td>97.8</td>
<td>98.7</td>
</tr>
</tbody>
</table>
I.1 Decimation ⇒ Hamiltonian Cycles

- Time complexity
  - decimation procedure $\sim O(M^2)$
  
  e.g. $q_c(k) = \delta_{k,c}: c = 3(+) , 4(\times), 5(\ast)$
I.1 Decimation ⇒ Hamiltonian Cycles

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

![Graph showing the relationship between no. BP-steps and M. The slope is labeled as approximately 0.23.](image)

slope $\approx 0.23$
I.1 Decimation ⇒ Hamiltonian Cycles

- Time complexity
  - decimation procedure $\sim O(M^2)$
  - 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 ⇒ Hamiltonian Cycles

- Time complexity
  - decimation procedure \( \sim O(M^2) \)
  - 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 ⇒ CC → HC
1.2 Markov Chain Monte Carlo Sampling

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

→ Ergodic? Convergence time?

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

$$W(S \rightarrow S') \text{Prob}[S] = W(S' \rightarrow S) \text{Prob}[S']$$
1.2 Markov Chain Monte Carlo Sampling

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

→ Ergodic? Convergence time?

→ Determine appropriate transitions \( S \to S' \), and transition rates

\[
W(S \to S'') : \text{e.g. by means of detailed balance:}
\]

\[
W(S \to S') \text{Prob}[S] = W(S' \to S) \text{Prob}[S']
\]

\[
\text{Prob}[S] = \begin{cases} 
0 & \text{if } S \text{ is not a cycle} \\
 f(u) & \text{if } S \text{ is a cycle} 
\end{cases} = \frac{1}{Z} u \sum_i S_i \prod_i f_i(S_i)
\]
I.2 Markov Chain Monte Carlo Sampling

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

→ Ergodic? Convergence time?

→ Determine appropriate transitions $S \to S'$, and transition rates $W(S \to S')$: e.g. by means of detailed balance:

\[ W(S \to S') \text{Prob}[S] = W(S' \to S) \text{Prob}[S'] \]

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

$n_S = \text{number of disjoint paths of configuration } S$

$\eta \in [0, 1)$

\[
\tilde{f}_i(S_i) = \begin{cases} 
1 & \text{if } \sum_{l \in \partial_i} S_l \in \{0, 2\} \\
\epsilon \in [0, 1] & \text{if } \sum_{l \in \partial_i} S_l = 1 \\
0 & \text{otherwise}
\end{cases}
\]
I.2 Monte Carlo ⇒ Hamiltonian Cycles

• Success rate:
  - Regular graphs of size $N = 100, 200, 400, 800$ : 100%
  - Bimodal graphs $(q_{3,4}^{0.5}, q_{3,5}^{0.5}, q_{4,5}^{0.5})$ of size
    $N = 100, 200, 400, 800$ : 100% → Confirmation of Wormald’s conjecture on non-regular graphs
I.2 Monte Carlo ⇒ Hamiltonian Cycles

- Succes rate:
  - Regular graphs of size $N = 100, 200, 400, 800$ : 100%
  - Bimodal graphs $(q_{3,4}^{0.5}, q_{3,5}^{0.5}, q_{4,5}^{0.5})$ of size
    $\quad N = 100, 200, 400, 800$ : 100% → Confirmation of Wormald’s conjecture on non-regular graphs

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

- Diagram showing the number of moves vs. $N$ for different values of $q_{3,4}^{0.5}$, $q_{3,5}^{0.5}$, and $q_{4,5}^{0.5}$.
Comparison

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

<table>
<thead>
<tr>
<th></th>
<th>BP</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>versatile</td>
<td>- very parameter sensitive</td>
</tr>
<tr>
<td>+</td>
<td>polynomial in $N$</td>
<td>- exponential in $N$</td>
</tr>
<tr>
<td>-</td>
<td>no guarantee</td>
<td>+ more reliable</td>
</tr>
</tbody>
</table>
Comparison

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

<table>
<thead>
<tr>
<th>BP</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ versatile</td>
<td>- very parameter sensitive</td>
</tr>
<tr>
<td>+ polynomial in $N$</td>
<td>- exponential in $N$</td>
</tr>
<tr>
<td>- no guarantee</td>
<td>+ more reliable</td>
</tr>
</tbody>
</table>

→ 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 optimized 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 $D(i) = |\partial i|$

(+) 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 \( D(i) = |\partial i| \)

(+) 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 $D(i) = |\partial i|$  

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

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

(+) iterative algorithm, emulates behavior of a Random Walk
II. Vertex (and Edge) Ranking

Ranking is an objective \textit{(topology based)} measure of importance of the vertices of a graph.

\textbf{Degree} \( D(i) = |\partial i| \)

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

\textbf{PageRank} \( P(i) \propto d \sum_{j \in \partial^+_i} \frac{P(j)}{d_j} \)

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

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

(+) 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** $D(i) = |\partial i|$
- (+) easy to compute
- (-) very rough measure

**PageRank** $P(i) \propto d \sum_{j \in \partial^+_i} \frac{P(j)}{d_j}$
- (+) iterative algorithm, emulates behavior of a Random Walk

**Betweenness Centrality** $B(i) = \sum_{k,l(\neq i) \in V} \frac{\sigma_{k,l}(i)}{\sigma_{k,l}}$
- (+) based on shortest paths
- (-) time requirements $\sim \mathcal{O}(NM)$

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

for $\text{Prob}[\mathcal{S}] = \frac{1}{Z} \prod_{l} (r_l)^{S_l} \prod_{i} f_i(S_i)$
Directed Small World Network
### Directed Small World Network

![Directed Small World Network Diagram](image)

<table>
<thead>
<tr>
<th>R</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>i</td>
<td>$P(i)$</td>
<td>$L(i)$</td>
<td>$B(i)$</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0.94</td>
<td>0.92</td>
<td>0.68</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.91</td>
<td>0.90</td>
<td>0.58</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.90</td>
<td>0.89</td>
<td>0.55</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0.84</td>
<td>0.87</td>
<td>0.50</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>0.78</td>
<td>0.84</td>
<td>0.45</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0.71</td>
<td>0.83</td>
<td>0.45</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.53</td>
<td>0.82</td>
<td>0.43</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>0.53</td>
<td>0.82</td>
<td>0.42</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.49</td>
<td>0.81</td>
<td>0.39</td>
</tr>
</tbody>
</table>
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).
  → Deeper investigation of the level of approximation of BP.
  → Improve MC by finding optimal parameters in automated way.
  → Find loops or paths of intermediate length.
  → Investigate real-world networks (scale free, weighted).
  → Consider a Potts-like configuration space.