# PageRank Optimization in Polynomial Time by Stochastic Shortest Path Reformulation

**Balázs Csanád Csáji**

University of Melbourne

joint work with:

R. M. Jungers & V. D. Blondel

21st International Conference on Algorithmic Learning Theory, Canberra, Australia, 2010

# Measuring Importance

- PageRank is a way to measure the importance of nodes in digraphs.
- The PageRank of a node can be interpreted as the average portion of time spent at the node by an infinite uniform random walk.
- The PageRank vector of a digraph is defined as the stationary distribution of an associated homogeneous Markov chain.
- PageRank was introduced by S. Brin and L. Page and is traditionally applied for ordering web-search results, e.g., it is a part of Google.
- It also has many other applications, for example, in bibliometrics, ecosystems, spam detection, web-crawling, semantic networks, relational databases and natural language processing.

# PageRank Optimization

- It is of natural interest to optimize the PageRank of a node.
- A webmaster could be, e.g., interested in increasing the PageRank of his website by suitably placing hyperlinks, e.g., advertisements, alliances.
- Sometimes we only have partial information of the graph structure, but still want to estimate the PageRank of a node in presence of these hidden, fragile links, e.g., the max/min possible PageRank of a node.
- We analyze the problem of optimizing the PageRank of a node by selecting edges from a subset of edges which are under our control.
- We show that this problem is essentially a stochastic shortest path problem and it can be solved in polynomial time.

# Overview

# PageRank: Strongly Connected Case

- Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges.
- First, assume that $\mathcal{G}$ is strongly connected.
- Then, $A$, the adjacency matrix of $\mathcal{G}$ is irreducible.
- Define a Markov chain on the graph by $P \triangleq \left(D_A^{-1} A\right)^{\mathrm{T}}$, where $D_A$ is diagonal and $(D_A)_{ii} \triangleq deg(i)$, the out-degree of node $i$.
- The PageRank vector of $\mathcal{G}$ is defined as the stationary distribution

$$\boxed{P\,\boldsymbol{\pi} = \boldsymbol{\pi}}$$

where $\boldsymbol{\pi}$ is non-negative and $\boldsymbol{\pi}^{\mathrm{T}}\boldsymbol{e} = 1$, with $\boldsymbol{e} = \langle 1, \ldots, 1\rangle^{\mathrm{T}}$.

- Vector $\boldsymbol{\pi}$ always exists and it is unique (Perron-Frobenius theorem).

# PageRank: General Case

- In the general case, there may be dangling nodes in graph $\mathcal{G}$ that do not have any outgoing edges.
- Assume that we handled them and all nodes have at least one out-link.
- Define $P$ as before. It may not have a unique stationary distribution.
- Thus, vector $\boldsymbol{\pi}$ is now the stationary distribution of the Google matrix

$$\boxed{G \triangleq (1-c)\,P + c\,\boldsymbol{z}\boldsymbol{e}^{\mathrm{T}}}$$

  where $\boldsymbol{z}$ is a positive personalization vector satisfying $\boldsymbol{z}^{\mathrm{T}}\boldsymbol{e} = 1$, and $c \in (0, 1)$ is a damping constant.
- The Markov chain defined by $G$ is ergodic that is irreducible and aperiodic, hence, its stationary distribution uniquely exists.

# PageRank Computation

- The PageRank of a node $i$ can be interpreted as the "importance" of $i$.
- Therefore, $\boldsymbol{\pi}$ defines a linear order on the nodes of the graph by treating $i \leq j$ if and only if $\boldsymbol{\pi}(i) \leq \boldsymbol{\pi}(j)$.
- The PageRank vector can be iteratively approximated by

  $$x_{n+1} \triangleq G x_n,$$

  starting from an arbitrary stochastic vector.
- It can also be directly computed by a matrix inversion

  $$\boldsymbol{\pi} = c\,(I - (1-c)P)^{-1}\boldsymbol{z},$$

  where $I$ denotes an $n \times n$ identity matrix.

# PageRank Optimization

- We are given a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a node $v \in \mathcal{V}$ and a set $\mathcal{F} \subseteq \mathcal{E}$ corresponding to those edges which are under our control.
- We can choose which edges in $\mathcal{F}$ are present and which are absent, but the edges in $\mathcal{E} \setminus \mathcal{F}$ are fixed, they must exist in the graph.
- $\mathcal{F}_+ \subseteq \mathcal{F}$ is a configuration: $\mathcal{F}_+$ determines those edges that we add to the graph, while $\mathcal{F}_- = \mathcal{F} \setminus \mathcal{F}_+$ denotes those edges which we remove.
- The PageRank of node $v$ under the $\mathcal{F}_+$ configuration is the PageRank of $v$ with respect to the graph $\mathcal{G}_0 = (\mathcal{V}, \mathcal{E} \setminus \mathcal{F}_-)$.
- Main question: how should we configure the fragile links, in order to maximize (or minimize) the PageRank of a given node $v$?

# Max-PageRank Problem

- The resulting combinatorial optimization problem can be summarized as

THE MAX-PAGERANK PROBLEM

| | |
|---|---|
| Instance: | A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a node $v \in \mathcal{V}$ and a set of controllable edges $\mathcal{F} \subseteq \mathcal{E}$. |
| Optional: | A damping constant $c \in (0, 1)$ and a stochastic personalization vector $z$. |
| Task: | Compute the maximum possible PageRank of $v$ by changing the edges in $\mathcal{F}$ and provide a configuration of edges in $\mathcal{F}$ for which the maximum is taken. |

- Our main contribution is that we show that Max-PageRank can efficiently (in polynomial time) reduced to a stochastic shortest path problem.

- Therefore, it can be solved in polynomial time and it is well-suited for reinforcement learning algorithms.

# Stochastic Shortest Path Problems

A stochastic shortest path (SSP) problem is defined as

- $\mathbb{S} = \{1, \ldots, n, n+1\}$ is a finite set of states
- $\mathbb{U}$ is a finite set of control actions
- $\mathcal{U} : \mathbb{S} \to \mathcal{P}(\mathbb{U})$ is an action constraint function
- $p : \mathbb{S} \times \mathbb{U} \to \Delta(\mathbb{S})$ is the transition function, $p(j \,|\, i, u)$ denotes the probability of arriving at state $j$ after taking action $u \in \mathcal{U}(i)$ in state $i$
- $g : \mathbb{S} \times \mathbb{U} \times \mathbb{S} \to \mathbb{R}$ is an immediate cost (or reward) function
- $\tau = n + 1$ is the target state; $\forall u$: $g(\tau, u, \tau) = 0$ and $p(\tau \,|\, \tau, u) = 1$

An SSP problem is an undiscounted Markov decision process (MDP) with an absorbing, cost-free termination state.

# Definitions and Notations

- A control policy is function from states to actions, $\mu : \mathbb{S} \to \mathbb{U}$.
- Policy $\mu$ is proper if, using $\mu$, $\tau$ can be reached from all states w.p.1.
- The cost-to-go function of policy $\mu$, $J^\mu : \mathbb{S} \to \mathbb{R}$ is defined as

$$J^\mu(i) \triangleq \lim_{k\to\infty} \mathbb{E}_\mu \left[ \sum_{t=0}^{k-1} g(i_t, u_t, i_{t+1}) \,\middle|\, i_0 = i \right]$$

  for all states $i$, where $i_t$ and $u_t$ are random variables representing the state and the action taken at time $t$, respectively.

- The Bellman optimality equation is $TJ^* = J^*$ where

$$(TJ)(i) \triangleq \min_{u\in\mathcal{U}(i)} \sum_{j=1}^{n+1} p(j \,|\, i, u) \Big[ g(i, u, j) + J(j) \Big]$$

# Linear Programming

- The optimal cost-to-go, $J^*(1), \dots, J^*(n)$, solves the following linear program in variables $x_1, \dots, x_n$:

$$\begin{array}{|lc|}\hline \text{maximize} & \displaystyle\sum_{i=1}^{n} x_i \\ \text{subject to} & \displaystyle x_i \leq \sum_{j=1}^{n+1} p(j \mid i, u)\Big[\, g(i,u,j) + x_j \Big] \\ \hline \end{array}$$

  for all actions $u \in \mathcal{U}(i)$; note that $x_{n+1}$ is fixed at zero.

- Hence, SSPs can be solved in polynomial time in the number of states, the number of actions and the binary size of the input.

- Moreover, SSP problems (along with other finite MDPs) are P-complete.

# Expected First Return Time

- Let $(X_0, X_1, \dots)$ denote a Markov chain defined on a finite set $\Omega$.
- The expected first return time of state $i \in \Omega$ is

$$\boldsymbol{\varphi}(i) \triangleq \mathbb{E}\left[\inf\{\, t \geq 1 : X_t = i \,\} \mid X_0 = i\,\right]$$

- If state $i$ is recurrent, $\boldsymbol{\varphi}(i)$ is finite; and if the chain is irreducible,

$$\boldsymbol{\pi}(i) = \frac{1}{\boldsymbol{\varphi}(i)},$$

  for all states $i$, where $\boldsymbol{\pi}$ is the stationary distribution of the chain.
- Thus, $\boldsymbol{\pi}(i)$ can be interpreted as the average portion of time spent in $i$.
- Moreover, maximizing [minimizing] the PageRank of a node is equivalent to minimizing [maximizing] the expected first return time to this node.

# Assumptions

- First, we start analyzing Max-PageRank without damping, $c = 0$.
- We will apply two assumptions, in order to simplify the presentation:

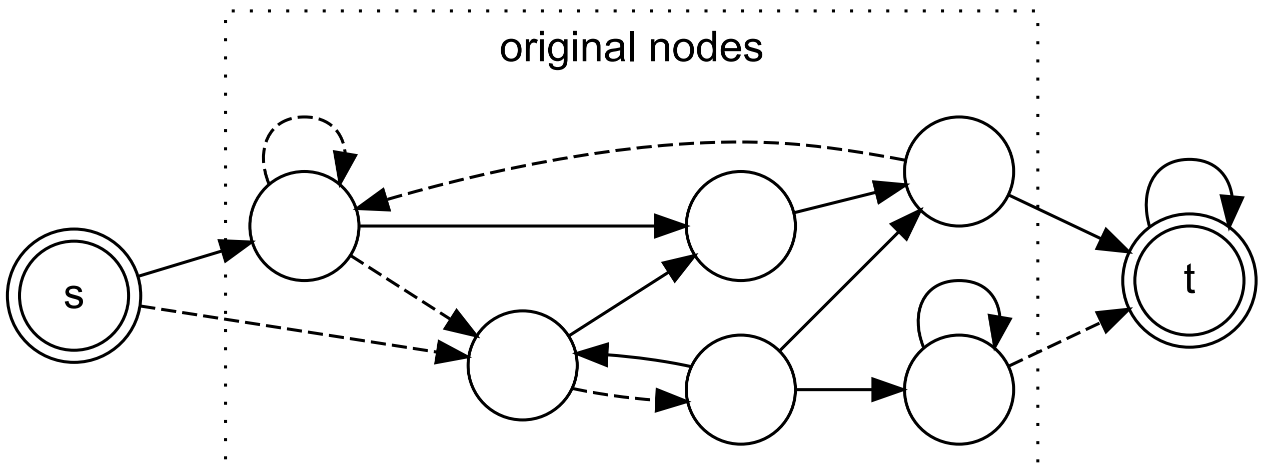
(AD) Dangling Nodes Assumption : We assume that there is a fixed (not fragile) outgoing edge from each node. It guarantees that there are no dangling nodes and there are no nodes with only fragile links.

(AR) Reachability Assumption : We also assume that for at least one configuration of fragile links we have a unichain process and node $v$ is recurrent. It is always true in case of damping.

- In SSP terminology (AR) assures that there is at least one proper policy.
- Note that these assumptions are not needed for the final result.

# Simple SSP Formulation

- We are going to reduce Max-PageRank to an SSP problem.
- The states of the MDP are the nodes of the graph, except for $v$ which we "split" into $v_s$ and $v_t$, a starting and a target state, respectively.
- State $v_s$ has all the outgoing edges of $v$ (both fixed and fragile)
- State $v_t$ has all the incoming edges of $v$ and a self-loop



# Simple SSP Formulation

- An action in state $i$ is to select a subset of fragile links (starting from $i$) which we "turn on" (activate).

- The transition probability from state $i$ to (a neighboring) state $j$ is $p(j \mid i, u) \triangleq 1/(a_i + b_i(u))$ if in $i$ there are $a_i \geq 1$ fixed outgoing edges and we have activated $b_i(u) \geq 0$ fragile links.

- The immediate-cost function is for all states $i$, $j$ and action $u$ is

$$g(i,u,j) \triangleq \begin{cases} 0 & \text{if } \; i = v_t \\ 1 & \text{otherwise} \end{cases}$$

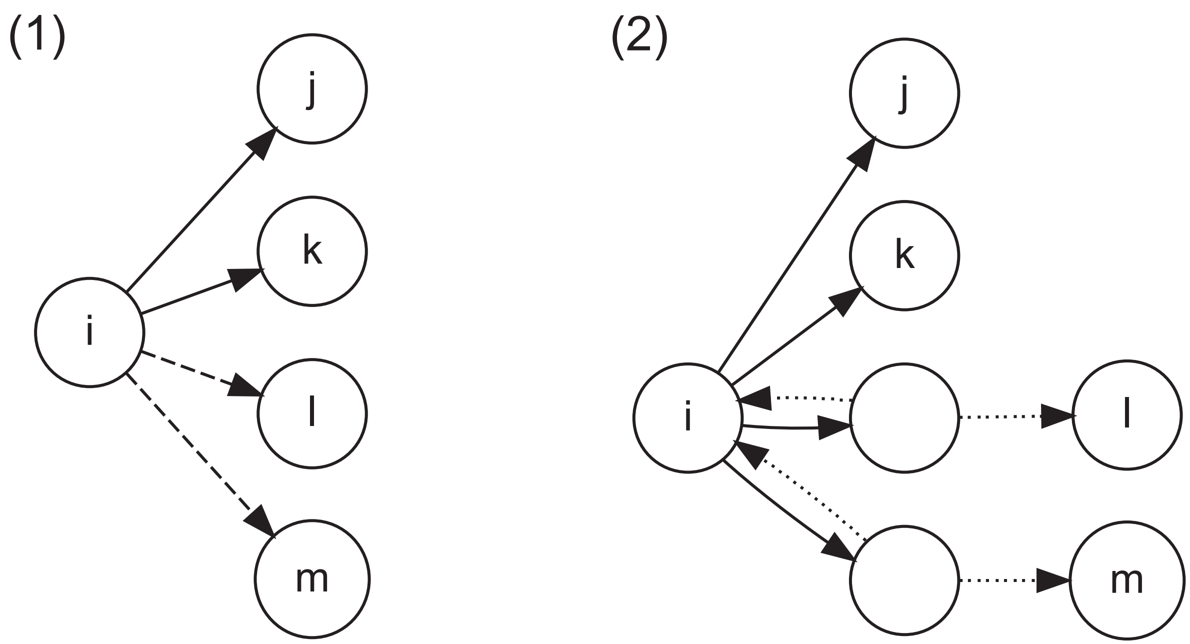
- Note that $J^{\mu}(v_s)$ is the expected first return time to node $v$ under $\mu$.

- Therefore, the maximum PageRank $v$ can have is $\boldsymbol{\pi}(v) = 1/J^{*}(v_s)$.

- But, this reduction is not polynomial, because of the action space.

# Reducing the Action Space

- The key idea is to introduce an auxiliary state, $f_{ij}$, for each fragile link.
- In each $f_{ij}$ there are two actions "on" and "off", these lead with probability one to node $j$ ("on") and back to node $i$ ("off"), respectively.
- The original fragile links starting from $i$ are changed to fixed ones.



# Refined SSP Formulation

- Claim: the transition probabilities between the original vertices of the graph are not effected by this reformulation.
- The immediate cost function should be modified, as well, not to count steps in the auxiliary states. Thus, for all states $i$, $j$, $l$ and action $u$

$$g(i,u,j) \triangleq \begin{cases} 0 & \text{if } i = v_t \text{ or } j = f_{il} \;\; or \;\; u = \text{``off''} \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

- The number of states of this formulation is $n + d + 1$, where $n$ is the number of nodes of the graph and $d$ is the number of fragile links.
- Moreover, the maximum number of allowed actions per state is $2$.
- (AD) & (AR) $\Rightarrow$ Max-PageRank can be solved in polynomial time.

# Linear Programming Formulation

- The resulted SSP problem can be reformulated as a linear program

$$\text{maximize} \quad \sum_{i \in \mathcal{V}} x_i + \sum_{(i,j) \in \mathcal{F}} x_{ij} \tag{2a}$$

$$\text{subject to} \quad x_{ij} \leq x_i\,, \quad \text{and} \quad x_{ij} \leq x_j + 1\,, \quad \text{and} \tag{2b}$$

$$x_i \leq \frac{1}{deg(i)} \Bigg[ \sum_{(i,j) \in \mathcal{E} \setminus \mathcal{F}} (x_j + 1) + \sum_{(i,j) \in \mathcal{F}} x_{ij} \Bigg], \tag{2c}$$

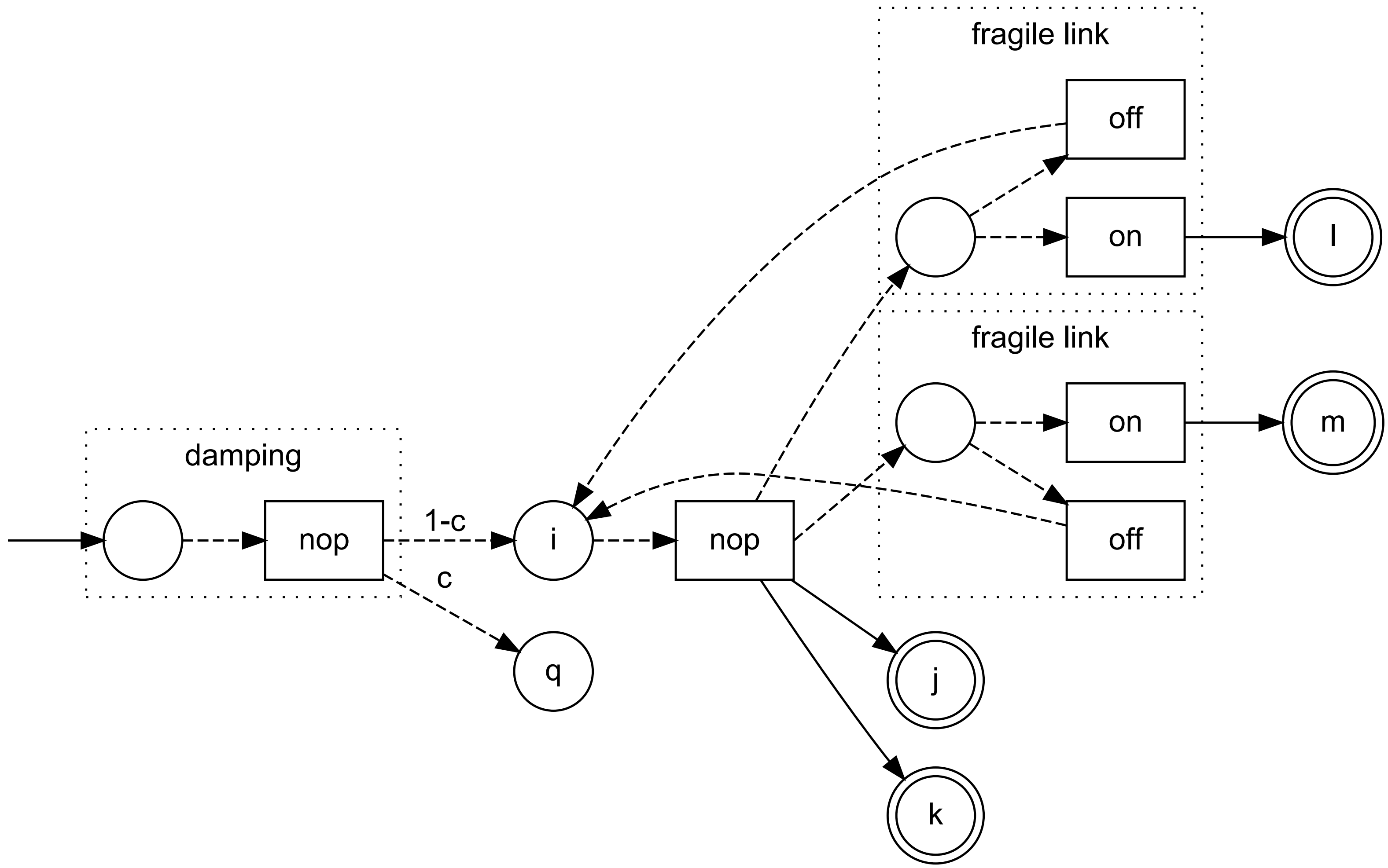
for all $i \in \mathcal{V} \setminus \{v_t\}$ and $(i,j) \in \mathcal{F}$.

- Notations: $x_i$ is the cost-to-go of state $i$, $x_{ij}$ relates to the auxiliary states of the fragile edges, and $deg(\cdot)$ denotes out-degree.
- Claim: (AD) is not necessary, dangling nodes can be handled.

# Damping and Personalization

- We now consider the general case with damping, $c \in (0, 1)$.
- Interpretation of damping: in each step we continue the random walk with probability $1 - c$ and we restart it, "zapping", with probability $c$.
- In case of zapping, the distribution of the new state is $\boldsymbol{z}$, the personalization vector.
- Again, auxiliary states are introduced to the previous solution.
- Auxiliary states $h_i$ are introduced for each state $i$, for damping.
- A global teleportation node $q$ is also introduced for personalization.
- The modification of transitions and costs is straightforward.

# Damping and Personalization



# Linear Programming Formulation

- The linear programming formulation in the general case is

$$\text{maximize} \quad \sum_{i \in \mathcal{V}} (x_i + \hat{x}_i) + \sum_{(i,j) \in \mathcal{F}} x_{ij} + x_q \tag{3a}$$

$$\text{subject to} \quad x_{ij} \leq \hat{x}_j + 1\,, \qquad \text{and} \qquad \hat{x}_i \leq (1 - c)\, x_i + c\, x_q\,, \tag{3b}$$

$$x_{ij} \leq x_i\,, \qquad \text{and} \qquad x_q \leq \sum_{i \in \mathcal{V}} \hat{z}_i\, (\hat{x}_i + 1)\,, \tag{3c}$$

$$x_i \leq \frac{1}{deg(i)} \Bigg[ \sum_{(i,j) \in \mathcal{E} \setminus \mathcal{F}} (\hat{x}_j + 1) + \sum_{(i,j) \in \mathcal{F}} x_{ij} \Bigg], \tag{3d}$$

for all $i \in \mathcal{V} \setminus \{v_t\}$ and $(i, j) \in \mathcal{F}$, where $\hat{z}_i = p(\, h_i \,|\, q)$, $\hat{x}_i$ denotes the cost-to-go of state $h_i$ and $x_q$ is the value of the teleportation state, $q$.

# Main Theorem

- We can summarize the results of the SSP reduction as

  **Theorem 1.** *The* MAX-PAGERANK PROBLEM *can be solved in polynomial time (under the Turing model of computation) even if the damping constant and the personalization vector are part of the input.*

- Note that assumptions (AD) and (AR) are not needed for this theorem,
- The method is also independent on how dangling nodes are handled.

# Exclusive Constraints

- The problem with exclusive constraints between the fragile links:

THE MAX-PAGERANK PROBLEM UNDER EXCLUSIVE CONSTRAINTS

| | |
|---|---|
| Instance: | A digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a node $v \in \mathcal{V}$, a set of controllable edges $\mathcal{F} \subseteq \mathcal{E}$ and a set $\mathcal{C} \subseteq \mathcal{F} \times \mathcal{F}$ of those edge-pairs that cannot be activated together. A damping constant $c \in (0, 1)$ and a stochastic personalization vector $z$. |
| Task: | Compute the maximum possible PageRank of $v$ by activating edges in $\mathcal{F}$ and provide a configuration of edges in $\mathcal{F}$ for which the maximum is taken. |

- Claim: the decision version of this problem is NP-complete.
- The proof is based on reducing $3$SAT to this problem.

# Summary and Conclusion

- The importance of nodes is often measured by their PageRank.
- The Max-PageRank problem asks for optimizing the PageRank of a node by adding or removing edges from a given subset of fragile links.
- We showed that Max-PageRank can be effectively reduced to a stochastic shortest path problem.
- It not only proves that it can be computed in polynomial time, but also shows that it is well-suited for reinforcement learning algorithms.
- The damping constant and the personalization vector can be part of the input and it does not matter how dangling nodes are handled.
- Our approach can be generalized to weighted graphs, as well.
- A constrained version of Max-PageRank is, however, already NP-hard.

# Thank you for your attention!