# Personalized Hitting Time for Informative Trust Mechanisms Despite Sybils

# Personalized Hitting Time for Informative Trust Mechanisms Despite Sybils

Brandon K. Liu
Paulson School of Engineering
Harvard University
brandonliu@post.harvard.edu

David C. Parkes
Paulson School of Engineering
Harvard University
parkes@eecs.harvard.edu

Sven Seuken
Department of Informatics
University of Zurich
seuken@ifi.uzh.ch

## ABSTRACT

Informative and scalable trust mechanisms that are robust to manipulation by strategic agents are a critical component of multi-agent systems. While the global hitting time mechanism (GHT) introduced by Hopcroft and Sheldon [9] is more robust to manipulation than PageRank, strategic agents can still benefit significantly under GHT by performing sybil attacks. In this paper, we introduce the *personalized hitting time mechanism* (PHT), which we show to be significantly more robust to sybil attacks than GHT. Specifically, if an agent has already cut all of its outlinks under PHT (which only leads to a negligible benefit), then adding sybils leads to no additional benefit. We provide an experimental analysis which demonstrates that, in the presence of strategic agents that create sybils, PHT dominates GHT (as well as PageRank and personalized PageRank) in terms of informativeness. We find the large dominance of PHT over GHT particularly surprising given the small difference between the two mechanisms. Finally, we provide a Monte Carlo algorithm to compute approximate PHT scores at scale, and we show that PHT retains its robustness to manipulation when used with approximate scores.

## CCS Concepts

**•Applied computing → Economics; •Information systems → Reputation systems;**

## Keywords

Trust Mechanisms; Reputation Mechanisms; Mechanism Design; Sybils; Informativeness

## INTRODUCTION

With the continued emergence of market-based systems (consider Uber and Airbnb) we get closer to the vision of agent-mediated commerce [15], with automated software agents negotiating and trading on behalf of a network of individuals who know very little about each other. Problems of trust are of the utmost importance: in identifying trustworthy counter-parties with whom to transact (who to take a ride with) and with the knock-on effect of promoting trustworthy behavior (clean cars, direct routes).

A *trust mechanism* takes reports about the experiences agents have with others, aggregates this information, and shares it in some form with the agents. An effective trust mechanism must be scalable and robust against manipulation in the presence of strategic agents who are willing to misreport trust to improve their own score or reduce the score of others. Ultimately we care about a mechanism that is informative, able to generate trust scores that discriminate between low quality and high quality agents. Specifically, we care about informativeness in the presence of strategic agents.

Some trust mechanisms are *global mechanisms*, which assign a single trust score to each agent, reflecting an aggregate view of the agent's trustworthiness based on all reports. Other trust mechanisms are *personalized* and may assign the same agent different trust scores depending on the viewpoint taken; e.g., agent $v_k$'s score from the perspective of $v_i$ may not be the same as $v_k$'s score from the perspective of $v_j$. We model this as a graph-theoretic problem: agents are nodes; agents' reports are weighted directed edges indicating trust; and trust mechanisms are graph-based algorithms.

The PageRank algorithm is a global mechanism originally developed to identify reputable web pages [14]. PageRank is known to be easily manipulated by *sybil attacks* [4], in which an agent introduces and controls the reports of a number of fake agents (the sybils). PageRank, for example, is vulnerable to a "two-loop attack," in which an agent indicates trust in its sybils, and its sybils indicate trust in the agent. Sybil attacks have been a concern in many multi-agent systems (e.g., in auction-based systems [19]).

In response to this concern, Hopcroft and Sheldon [9] (HS) introduced the *global hitting time mechanism* (GHT). GHT is closely related to PageRank. In fact, a transformed GHT score for an agent $v_k$ can be computed via PageRank on a modified graph with all of $v_k$'s out-edges removed. This difference is crucial. Whereas a PageRank score for agent $v_k$ may depend on its own as well as others reports, a GHT score is independent of an agent's own reports. Although this removes the two-loop attack, GHT remains vulnerable to sybils through a "restart-capture attack."[1] HS point this out, stating GHT can be "heavily swayed" in the extreme of a large number of sybils. These authors also suggest the idea of "limiting the restart probability granted to new nodes" and point to personalized variations on PageRank [7], as well as variations of PageRank that use pre-trusted nodes for restart [8] (see also Kamvar et al. [10]).

HS provide a limiting analysis, pointing out that sybils can boost a trust score by a large additive amount, reflecting the probability of restart. But HS leave open the question as to the practical implication of restart recapture in GHT, especially its effectiveness relative to other manipulations, its consequences for GHT relative

**Appears in:** *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016),*
*J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.),*
*May 9–13, 2016, Singapore.*

[1] Restart-capture refers to the role of sybils in capturing probability when the random walk used to define GHT scores restarts and jumps to a new node selected at random.

to other mechanisms, and its effect on the informativeness of GHT. This paper closes this gap.

We study the *personalized hitting time* (PHT) mechanism, which does not appear to have been discussed or studied in the prior literature. We make the following contributions:

- We prove that the optimal manipulation strategy under PHT only involves cutting outlinks, but does not involve adding sybils. In comparison, the optimal manipulation strategies under PageRank, GHT and a personalized variation of PageRank all require sybils.
- Through a detailed experimental analysis, we quantify the extent to which PHT, GHT, PageRank, and personalized PageRank are vulnerable to sybil attacks. We demonstrate that sybil attacks are especially powerful in global mechanisms (GHT and PageRank) and also effective in personalized PageRank. We show that PHT is very robust to sybil attacks, which translates into a large improvement in informativeness relative to GHT and other mechanisms. The dominance of PHT over GHT is particularly surprising given the small difference between the two mechanisms.
- We provide a Monte Carlo algorithm to compute PHT scores and bound the error as a function of the number of samples. Furthermore, we show that an approximate version of the PHT mechanism retains robustness against sybil manipulations by strategic agents.

## Related Work

Trust mechanisms have been proposed for various applications, including the efficient identification of web spam [8], searching for trusted web pages on the Internet [14], and identifying good peers [10, 12]. Yu et al. [22] give an extensive survey of the wide range of trust mechanisms and trust management systems in the literature.

Although Cheng and Friedman's [3] conception of a successful sybil manipulation differs slightly from our own (since they allow an agent to transact as a sybil), and although they only consider sybil attacks in isolation from other manipulations, their main result is relevant here— non-trivial global trust mechanisms are susceptible to sybil attacks while personalized trust mechanisms can be robust against sybils.

Hopcroft and Sheldon [9] introduce GHT and provide a theoretical treatment of its manipulation-resistance in regard to both dropping trust reports and using sybils. But as discussed above, they do not explore in simulation the remaining vulnerability of GHT to sybil attacks, study a personalized variation, or consider informativeness.

Tang et al. [18] show that max-flow and a shortest-path based mechanism are less informative than PageRank and GHT in the absence of strategic agents. Based on this, we restrict our attention to mechanisms based on random-walks in the style of PageRank and GHT.

Seuken et al. [17] have studied trust-based mechanisms for work accounting systems, where the goal is to incentivize users to contribute as much as they consume over time. Recently, Seuken and Parkes [16] have proven that it is impossible to design accounting mechanisms that use transitive trust (like PageRank and hitting time) and are are also sybil-proof. While their framework is slightly different from ours, their results demonstrate the challenges that using a notion of transitive trust imply for achieving robustness towards sybil attacks.

One way to achieve certain robustness against sybil attacks is to leverage external trust information. Unlike methods such as SybilGuard [21] and SybilLimit [20] that assume access to additional information such as pre-existing social links on social networks, we limit inputs to voluntary reports from agents.

## PRELIMINARIES

Let $V$ denote a set of agents, with cardinality $|V| = n$, and let $v_i, v_j \in V$ denote individual agents. In our model, agents engage in transactions with each other. One agent is the initiator and one is the provider. Transactions are risky and may succeed or fail. Each agent $v_i$ has a (latent) *type* $\theta_i \in [0, 1]$. This defines how trustworthy the agent is. A transaction initiated by $v_j$ with provider $v_i$ succeeds with probability $\theta_i$, depending on the type of the provider.

At any point in time, let $V_i$ denote the set of agents with whom agent $v_i$ has initiated a transaction. Following a transaction, we assume that an agent updates its belief about the type of the provider, e.g., by considering its prior and the success or failure of the transaction. We do not model this update explicitly for our theoretical model, but rather use $\beta_{ij} \in [0, 1]$ to denote agent $v_i$'s belief about an agent $v_j \in V_i$ with which it has transacted.

A trust mechanism takes a report from each agent. The report from agent $v_i$ is a claim (perhaps untruthful) about the trustworthiness of each agent $v_j \in \hat{V}_i \subseteq V_i$, where $\hat{V}_i$ is the set of other agents about which it makes reports. Let $\hat{w}_{ij} \in [0, 1]$ denote its report about agent $v_j$, and let $\hat{w}_i$ denote the complete set of reports from $v_i$. We assume that an agent can only make reports about agents with which it has initiated one or more transactions.

A *trust graph* $G = (V, E, \beta)$ is a weighted directed graph (digraph) with a set $V$ of vertices, a set $E$ of directed edges, and edge weights $\beta : E \mapsto \mathbb{R}_{\geq 0}$. We slightly abuse notation, using $V$ for both the set of agents and the set of vertices. A trust graph is constructed from a set of reports in the natural way, with agents corresponding to vertices, and a weighted directed edge from (the vertex corresponding to) agent $v_i$ to agent $v_j$ with weight $\hat{w}_{ij}$, for report $\hat{w}_{ij}$ from $v_i$ about $v_j$.

A *trust mechanism* takes a set of reports and returns a *trust score* $x_{ij} \in \mathbb{R}_{\geq 0}$ for every pair of agents $v_i, v_j \in V$ with $v_i \neq v_j$; this is the trust for $v_j$ from the viewpoint of $v_i$. In a global trust mechanism, we have $x_{ij} = x_{kj}$ for all agents $v_j$, and all $v_i, v_k$. In this case, we let $x_j$ denote the score of agent $v_j$. In a personalized trust mechanism, we may have $x_{ij} \neq x_{kj}$, with the score of agent $v_j$ depending on whether the viewpoint is that of $v_i$ or $v_k$.

The hitting time mechanisms are defined in terms of a random walk on a trust graph. This is a sequence of random variables $(X_0, X_1, \dots)$, where each $X_i \in V$, and

$$\mathbb{P}(X_{t+1} = v_j | X_t = v_i) = \frac{\beta_{ij}}{\sum_{(i,j') \in E} \beta_{ij'}}. \tag{1}$$

In defining this walk, a self-loop is added to any vertex that has no out-edges. Depending on the context, vertex $X_0$ may be fixed, or sampled according to some *restart distribution* $F_q$. The *hitting time* of vertex $v_j$ is $H(v_j) = \min\{t : X_t = v_i \mid X_0 \sim F_q\}$. An *$\alpha$-terminating random walk*, for some $\alpha \in [0, 1]$, is a finite random walk $(X_0, X_1, \dots, X_\tau)$ where the walk length is a random variable, $\tau \sim \text{Geom}(1 - \alpha)$. In effect, the random walk terminates with probability $\alpha$ in each step. Let $(X_t)_{t=0}^{\tau}$ denote the sequence of vertices visited by an $\alpha$-terminating random walk.

DEFINITION 1 (GLOBAL HITTING TIME MECH. (GHT)). *Given a reported trust graph, parameter $\alpha \in [0, 1]$, and restart distribution $F_q$, the global hitting-time score $x_{\text{GHT},j}$ of agent $v_j$ is the probability that an $\alpha$-terminating random walk initiated at a vertex sampled from $F_q$ visits $v_j$ before it restarts, i.e., $x_{\text{GHT},j} = \mathbb{P}(v_j \in (X_t)_{t=0}^{\tau} \mid X_0 \sim F_q)$.*

This is equivalent to defining the GHT score as $\mathbb{P}(H(v_j) \leq \tau)$, where $\tau$ is the distribution on path length before restart.

We assume that distribution $F_q$ puts some probability on sybils in our theoretical analysis, and take it to be uniform in our simulations. Given access to a fixed set of pre-trusted vertices, then restart can be limited to these vertices. Personalized hitting time is a special case, where the pre-trusted set is simply the agent itself.

DEFINITION 2 (PERSON. HITTING TIME MECH. (PHT)). *Given a reported trust graph and parameter* $\alpha \in [0, 1]$*, the personalized hitting time score* $x_{\mathrm{PHT},ij}$ *of agent* $v_j$ *as viewed from agent* $v_i$ *is the probability that an* $\alpha$*-terminating random walk that starts from* $v_i$ *visits* $v_j$ *before it restarts, i.e.,* $x_{\mathrm{PHT},ij} = \mathbb{P}(v_j \in (X_t)_{t=0}^{\tau} \mid X_0 = v_i)$.

This is equivalent to defining the PHT score for $v_j$ as viewed by $v_i$ as $\mathbb{P}(H(v_j) \leq \tau \mid X_0 = v_i)$, which denotes the probability that a random walk initiated at $v_i$ will visit $v_j$ before terminating.

DEFINITION 3 (PAGERANK MECHANISM). *Given a reported trust graph, parameter* $\alpha \in [0, 1]$*, and restart redistribution* $F_q$*, the global PageRank score* $x_{\mathrm{PR},j}$ *of agent* $v_j$ *is the steady-state probability that an* $\alpha$*-terminating random walk initiated at a vertex sampled from* $F_q$ *spends at* $v_j$.

DEFINITION 4 (PERSONALIZED PAGERANK MECH. (PPR)). *Given a reported trust graph and parameter* $\alpha \in [0, 1]$*, the personalized PageRank score* $x_{\mathrm{PPR},ij}$ *of agent* $v_j$ *as viewed from agent* $v_i$ *is the steady-state probability that an* $\alpha$*-terminating random walk that starts (and restarts) from* $v_i$ *spends at* $v_j$.

Collectively, we call these the *random-walk family* of trust mechanisms.

Motivated by the ubiquity of related schemes in social and economic platforms, we also adopt the *average-score mechanism* (AS) as a simple baseline. This computes $x_{\mathrm{AS},j}$ as the average of reports about $v_j$ by others. As we will show, AS turns out to have interesting properties. In particular, it is less manipulable than GHT and PageRank, and is relatively informative for scenarios where the agents are collectively well-informed about each other.

# THEORETICAL ANALYSIS

In this section, we analyze the robustness of PHT to manipulation and compare this with the other mechanisms.

A *strategic agent* is interested in increasing its own score and decreasing the scores of others (because such manipulations may lead to other agents initiating more transactions with this agent). One strategy is to *misreport* other agents as untrustworthy, which in the random-walk family of mechanisms is equivalent to dropping reports (and hence also known as *cutting outlinks*). A strategic agent $v_j$ can also execute a *sybil attack* by creating one or more sybils and arbitrary trust reports among itself and its sybils. Following previous research (e.g., Tang et al. [18]), we assume no trust reports from other agents to $v_j$'s sybils.[2]

We characterize the optimal combination of misreports and sybil attacks for each of our mechanisms.[3]

THEOREM 1. *The optimal manipulations for a strategic agent* $v_j$ *with access to one or more sybils are:*

- *GHT: Drop all trust reports about other agents. Add as many sybils as possible and have each sybil report trust 1 for* $v_j$.
- *PHT: Drop all trust reports about other agents. Do not add any sybils.*
- *PageRank: Drop all trust reports about other agents. Add as many sybils as possible with "two-loops" such that* $v_j$ *reports trust 1 for each sybil, and each sybil reports trust 1 for* $v_j$.
- *PPR: Drop all trust reports about other agents. Add one sybil and one two-loop with this sybil.*
- *AS: Report zero trust for every agent in set* $V_j$.[4] *Add as many sybils as possible and have each sybil report trust 1 for* $v_j$.

PROOF. • GHT: Dropping all trust reports to others is strictly dominant because a random walk (r.w.) at $v_j$ has already hit $v_j$ and an out-edge from $v_j$ can make it possible for the r.w. to also hit another agent. Thus, an out-edge leaves $v_j$'s own score unchanged while increasing the score of others. Consider adding a sybil $s$. A r.w. at $v_j$ has already hit $v_j$ and thus adding an edge to $s$ does not change the score of $v_j$. But sybil $s$ is useful for capturing a restarted r.w. and sending it in a single hop (the shortest possible) to $v_j$, thus improving the score of $v_j$. For this, $s$ has an out-edge (with weight 1) to $v_j$ and no other edges.[5]

• PHT: Consider $v_j$ and the viewpoint from some other agent $v_i$. The analysis in regard to dropping reports of other agents is the same as for GHT. In regard to a sybil $s$, it remains irrelevant to add an edge to $s$. In addition, $s$ no longer captures restart probability (except for trust scores as viewed from $s$), since all restart for the trust scores from the viewpoint of $v_i$ occurs at $v_i$.[6]

• PageRank: Adding a sybil $s$ with a two-loop from $v_j$ to $s$ and back to $v_j$ sends a r.w. at $v_j$ back to $v_j$ as quickly as possible (at least as effectively as if the r.w. visits some other vertex), increasing the visit probability at $v_j$. A sybil also captures restart probability, and a two-loop sends a r.w. that starts at $s$ to $v_j$ as quickly as possible. To maximize both effects it is optimal to have as many sybils as possible, each with two-loops to provide the shortest path possible back to $v_j$. Given one or more two-looped sybils, it is optimal for $v_j$ to drop all trust reports to other agents. An out-edge from $v_j$ to $v_i$ allows a r.w. at $v_j$ to also visit $v_i$, increasing the score of $v_i$. With one or more two-loops, there is no need to use $v_i$ to allow the r.w. to return to $v_j$ because $v_j$'s sybils already provide the shortest-possible return path.[7]

[2] In practice, obtaining a positive trust report for a sybil should be similarly costly as obtaining a trust report for oneself, but the latter should be at least as beneficial for one's reputation.

[3] The results for PageRank and PPR follow from Bianchini et al. [2] and Cheng and Friedman [4]. The results for GHT follow from Hopcroft and Sheldon [9]. The analysis for PHT is new. We provide a self-contained proof of the theorem for completeness.

[4] Reporting zero is different from dropping a report in AS because reports are averaged, and the absence of information is different from a report of zero.

[5] If it wasn't already optimal for $v_j$ to drop trust reports about others, there is an additional role for sybils in GHT. Beyond capturing restart probability, a sybil can divert a r.w. at $v_j$ away from other agents and reduce the score of these other agents. For this, $v_j$ would want to introduce a large number of sybils $s'$ with an edge from $v_j$ to $s'$, along with sybils $s$ that have an edge from $s$ to $v_j$. A similar observation can be made for PHT, where without dropping trust reports about others then it would be useful to introduce sybils $s'$ with an edge from $v_j$ to each $s'$.

[6] Similar to the analysis for GHT, if $v_j$ cannot drop reports, there is a role to play for sybils. However, a sybil attack without dropping reports is strictly less effective than dropping all reports. Consider an agent $v_k$ with outlinks to some sybils and without dropping any reports. A r.w. at $v_k$ has a non-zero probability of taking an outlink to another agent, whereas a r.w. at $v_j$ has a *zero* probability of doing so.

[7] Without sybils, it can sometimes be beneficial to keep an out-edge with an agent who reports trust in $v_j$. This depends on a balance

• PPR: The analysis is similar, but reveals that only one sybil is required. A single sybil $s$ with a two-loop from $v_j$ to $s$ and back to $v_j$ sends a r.w. at $v_j$ back to $v_j$ as quickly as possible (at least as effectively as if the r.w. visits some other vertex), increasing the visit probability at $v_j$. But now a sybil does not capture restart probability, and thus the second effect in PageRank is not present, and only one sybil is required. The rest of the analysis is unchanged, and dropping reports of other agents is optimal together with a two-loop with a sybil.

• AS: A trust report of 0 about another agent $v_i$ maximally decreases the trust score of $v_i$. Having a sybil $s$ report high trust for $v_j$ increases $v_j$'s score. A trust report by $v_j$ about $s$ has no effect on the score of $v_j$. □

Theorem 1 tells us that sybils do not bring new manipulation ability to a strategic agent in PHT who is already able to misreport trust about others. This property is unique to PHT amongst these mechanisms.

Sybils are useful for restart-capture in the global mechanisms, even when an agent is already dropping reports about others. Sybils with two-loops are useful in PageRank and PPR, and even when an agent is already dropping reports about others. Sybils are useful in AS.

Following Hopcroft and Sheldon [9], we can also quantify the effect of manipulation in PHT and compare this with GHT. Let $\rho = \sum_{s \in S} q(s)$, where $S$ is the set of sybils introduced by agent $v_j$ and $q(s)$ the restart probability (given by distribution $F_q$). This is the total probability captured by $v_j$'s sybils. Let $\mathit{infl}(j,k) = \mathbb{P}[H(v_j) < H(v_k) \leq \tau]$ denote the *influence* of $v_j$ on $v_k$ (where $H(v_j)$ and $H(v_k)$ may be correlated). Without sybils, the impact of strategic behavior on $v_k$ is equal to the change in influence of $v_j$ on $v_k$. For PHT, let $\mathit{infl}(j,k \mid i) = \mathbb{P}[H(v_j) < H(v_k) \leq \tau \mid X_0 = v_i]$ denote the *influence* of $v_j$ on $v_k$'s trust score, as determined from the viewpoint of $v_i$. We have

$$\begin{aligned}\mathit{infl}(j,k \mid i) &= \mathbb{P}[H(v_j) \leq \tau \mid X_0 = v_i] \\ &\quad \cdot \mathbb{P}[H(v_j) < H(v_k) \leq \tau \mid X_0 = v_i, H(v_j) \leq \tau] \\ &= x_{\mathrm{PHT},ij} \cdot \mathbb{P}[H(v_k) \leq \tau' \mid X_0 = v_j] \\ &= x_{\mathrm{PHT},ij} \cdot x_{\mathrm{PHT},jk}, \qquad (2)\end{aligned}$$

where $\tau' \sim \mathrm{Geom}(1-\alpha)$. The influence of $v_j$ on the score of $v_k$ in PHT, viewed from $v_i$, depends on the score $v_j$ gives to $v_k$, but dampened through the score $v_i$ gives to $v_j$. Let $x$ and $x'$ denote trust scores computed on the basis of true inputs and following manipulation by $v_j$, respectively.

THEOREM 2 (FOLLOWING HOPCROFT & SHELDON (2007)). *The effect on trust scores of manipulation by a strategic agent $v_j$ with access to sybils is:*

- *In GHT:*
  *(i) $x'_{\mathrm{GHT},j} \leq (1-\rho)x_{\mathrm{GHT},j} + \rho$, and*
  *(ii) $x'_{\mathrm{GHT},k} \geq (1-\rho)(x_{\mathrm{GHT},k} - \mathit{infl}(j,k))$, $\forall v_k \neq v_j$.*
- *In PHT: for any observer $v_i \neq v_j$,*
  *(i) $x'_{\mathrm{PHT},ij} = x_{\mathrm{PHT},ij}$, and*
  *(ii) $x'_{\mathrm{PHT},ik} \geq x_{\mathrm{PHT},ik} - \mathit{infl}(j,k \mid i)$, $\forall v_k \notin \{v_i, v_j\}$.*

The properties for GHT are stated as Theorem 4.10 (i) and (ii) in Hopcroft and Sheldon [9]. The properties for PHT follow from Corollary 4.7 (i) and (ii) in Hopcroft and Sheldon [9], using our Theorem 1 to reduce the properties of PHT with strategic behavior in the presence of sybils to the properties of GHT with strategic behavior in the absence of sybils.

PHT with sybils enjoys the same analytic results in regard to strategic behavior as those for GHT in a system where sybils are precluded by assumption. From properties (i), whereas an agent in GHT may be able to increase its own score by as much as $\rho$, it cannot change its own score in PHT. From properties (ii), whereas an agent in GHT may be able to both remove the effect of its influence on the score of another agent and further dampen this by a factor $(1-\rho)$, the second factor goes away in PHT.

# COMPUTING PHT TRUST SCORES

Large-scale applications of trust mechanisms require algorithms that can quickly compute trust scores. We will now first present an exact method for computing PHT, which is based on solving a system of linear equations that relate hitting time values to each other, but that runs in in $\mathcal{O}(n^4)$. In a second step we will then present a Monte Carlo approximation method.

## An Exact Algorithm for Computing PHT

ALGORITHM 1 (EXACT). *Given a weighted digraph $G$, let $\vec{x}(j)$ be the vector of PHT scores from every vertex to $v_j$, i.e., $\vec{x}_i(j) = x_{\mathrm{PHT},ij}$. Set parameter $\alpha$. Let $P$ denote the transition matrix of $G$. Solving the system of linear equations defined by*

$$\vec{x}(j) = (1-\alpha)P(j)\vec{x}(j) + \vec{e}_j \qquad (3)$$

*yields the values for $\vec{x}(j)$, where $P(j)$ is a modified transition matrix of $G$ where $P_{jk}(j) = 0$ for all $k$, and $\vec{e}_j$ is the standard basis vector.*

THEOREM 3. *The exact algorithm correctly computes the PHT scores in $\mathcal{O}(n^4)$ time.*

PROOF. Let $(X_t)_{t=0}^{\tau}$ be a finite sequence of random variables representing the sequence of states that are visited by an $\alpha$-terminating random walk on a weighted digraph $G$. Then the PHT score $x_{\mathrm{PHT},ij}$ is

$$x_{\mathrm{PHT},ij} = \mathbb{P}(v_j \in (X_t)_{t=0}^{\tau} \mid X_0 = v_i).$$

Conditioning on $X_1$ and expanding by the law of total probability yields

$$\sum_k \mathbb{P}(v_j \in (X_t)_{t=0}^{\tau} | X_0 = v_i, X_1 = v_k) \cdot \mathbb{P}(X_1 = v_k | X_0 = v_i).$$

When $i \neq j$, we can simplify the first term to $x_{\mathrm{PHT},kj}$ by the Markov property. The second term simplifies to the product of the survival probability $1-\alpha$ and the transition probability $P_{ik}$, where $P$ is the transition matrix of $G$. This yields

$$x_{\mathrm{PHT},ij} = (1-\alpha)\sum_k P_{ik} \cdot x_{\mathrm{PHT},kj}.$$

Let $P(j)$ be a modified transition matrix with the same entries as $P$ but with $P_{jk}(j) = 0$ for all $k$. We can now vectorize this expression for $x_{\mathrm{PHT},ij}$ with $j$ fixed, taking care to ensure that $x_{\mathrm{PHT},jj} = 1$. We make use of $P(j)$ and the standard basis vector $\vec{e}_j$ to produce the vectorization

$$\vec{x}(j) = (1-\alpha)P(j)\vec{x}(j) + \vec{e}_j.$$

To find all $n^2$ PHT scores, we must solve $n$ systems of these linear equations. As solving a system of $n$ linear equations in $n$ variables takes time $\mathcal{O}(n^3)$, the total algorithm takes $\mathcal{O}(n^4)$ time. □

between allowing a r.w. to visit this other agent, thus increasing the other agent's score, while also bringing the r.w. back to $v_j$, increasing $v_j$'s own score. A similar observation can be made for PPR.

Although each system of equations can be solved in parallel, and computation can be accelerated further by methods in parallel numerical linear algebra [6], the $\mathcal{O}(n^4)$ time complexity is likely to remain prohibitive in practical applications. For this reason, we will next present a significantly faster, albeit *approximate* approach for computing PHT.

## A Monte Carlo Method for Computing PHT

A simple Monte Carlo approximation would simulate $\alpha$-terminating random walks between every pair of vertices, and estimate $x_{\text{PHT},ij}$ as the proportion of walks starting at $v_i$ that reach $v_j$ before terminating. We improve on this, by introducing the following *multi-walk algorithm.*

ALGORITHM 2 (MULTI-WALK). *Given a trust graph $G = (V, E, \beta)$ (and $|V| = n$), initiate a total of $m$ $\alpha$-terminating random walks, with $m_i = m/n$ walks initiated at each vertex $v_i$. Let $Z_{ij}$ be the number of walks that include a subsequence that visits $v_i$ and subsequently $v_j$. Let $Y_i$ be the number of walks for which $v_i$ is visited (including those started at $v_i$). Estimate PHT scores by $\hat{x}_{\text{PHT},ij} = Z_{ij}/Y_i$.*

The multi-walk algorithm improves on the simple approach by (1) using a random walk (r.w.) from $v_i$ to estimate the trust score from $v_i$ to multiple other vertices; and (2) using a subsequence of a r.w. started at some vertex $v_i'$, to estimate the trust score from $v_i'$ to multiple other vertices. In order to avoid bias and keep samples independent, we must take care not to use a subwalk from a vertex that has already been visited on the current r.w.

THEOREM 4. *The estimate computed by the multi-walk algorithm is unbiased and consistent.*

PROOF. Consider all r.w.s that visit vertex $v_i$ (there are $Y_i$ of these). For such a random walk $(X_t)_{t=0}^{\tau}$, let $H_i$ denote the step (or hitting time) at which this first occurs. The estimator $\hat{x}_{\text{PHT},ij}$ is given by the proportion of these walks that subsequently visit $v_j$. The probability that the r.w. visits $v_j$ is $\mathbb{P}(v_j \in (X_t)_{t=H_i}^{\tau} | v_i \in (X_t)_{t=0}^{\tau})$. Using the time-homogeneous and Markovian property of the random walk, this is just $\mathbb{P}(v_j \in (W_t)_{t=0}^{\tau'} | W_0 = v_i)$, where $(W_t)_{t=0}^{\tau'} = (X_t)_{t=H_i}^{\tau}$ and $\tau' = \tau - H_i$. Due to the memoryless property of the geometric distribution, it also holds that $\tau' \sim \text{Geom}(1-\alpha)$. This is exactly the quantity $x_{\text{PHT},ij}$. Thus the proportion of random walks that succeed can be modeled as the average of $Y_i$ Bernoulli trials, each of which succeeds independently with probability $x_{\text{PHT},ij}$. Hence, $\hat{x}_{\text{PHT},ij} \sim \frac{1}{Y_i}\text{Binom}(Y_i, x_{\text{PHT},ij})$, and is an unbiased and consistent estimator for $x_{\text{PHT},ij}$. □

THEOREM 5. *For $\delta, \epsilon > 0$, obtaining an estimate $\hat{x}_{\text{PHT},ij}$ for which $\mathbb{P}(|Y_i(\hat{x}_{\text{PHT},ij} - x_{\text{PHT},ij})| \geq \epsilon Y_i x_{\text{PHT},ij}) \leq \delta$ requires $Y_i \geq \frac{3\ln(2/\delta)}{\epsilon^2 x_{\text{PHT},ij}}$, where $Y_i$ is the number of random walks that visit $v_i$.*

PROOF. The proof follows from a standard two-sided Chernoff bound analysis: Given $K$ independent random variables $X_1, \ldots, X_K$ with $\mathbb{E}[X_i] = \mu$ and $0 \leq X_i \leq 1$ for all $i$, let $\overline{X}$ be their mean. Then for any $\epsilon$, the Chernoff bound is given by:

$$\mathbb{P}(|\overline{X} - \mu| \geq \epsilon\mu) \leq 2\exp\left(-\frac{\epsilon^2}{3}K\mu\right). \tag{4}$$

Given $\epsilon$ and $0 < \delta < 1$, call $\overline{X}$ an $(\epsilon, \delta)$-approximation if $\mathbb{P}(|\overline{X} - \mu| \geq \epsilon\mu) \leq \delta$. For this, the Chernoff bound tells us we must have $K \geq \frac{3\ln(2/\delta)}{\epsilon^2\mu}$. Instantiating to the multi-walk algorithm (where the mean is $x_{\text{PHT},ij}$) yields the result. □

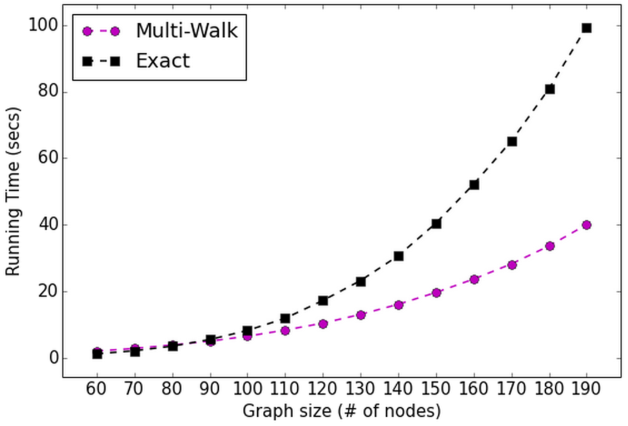


**Figure 1: Run-time of the exact and multi-walk algorithms for computing PHT scores as a function of graph size $n$.**

We now compare the run-time of the exact and approximate methods on preferential-attachment graphs [1], which are graphs that have a power-law degree distribution and replicate some of the statistical regularities seen in real-world social and economic networks [5]. Figure 1 provides the run-time of the exact and multi-walk algorithms on increasing graph sizes. We vary the number $n$ of vertices, and set the number of edges so that there are $n/10$ edges per vertex, with edge weights sampled $uniform(0, 1)$.[8] The multi-walk algorithm is faster for large graphs, with the cross-over occurring for systems with as few as 100 agents. Moreover, the multi-walk algorithm can be easily parallelized in a map-reduce environment.[9]

PROPOSITION 1. *The strategic properties of the approximate-PHT mechanism are the same as with the PHT mechanism.*

PROOF. In regard to the characterization (Theorem 1), the argument is based on the property of an individual r.w. and its role in determining trust scores, and does not depend on the number of r.w.s used (and thus the accuracy of the estimate). This also provides Theorem 2 (i) for approximate PHT. In regard to Theorem 2 (ii) for approximate PHT, the argument in Hopcroft and Sheldon [9] generalizes, just replacing $\mathit{infl}(i, k \mid j) = x_{\text{PHT},ji} \cdot x_{\text{PHT},ik}$ with $\mathit{infl}(i, k \mid j) = \hat{x}_{\text{PHT},ji} \cdot \hat{x}_{\text{PHT},ik}$. □

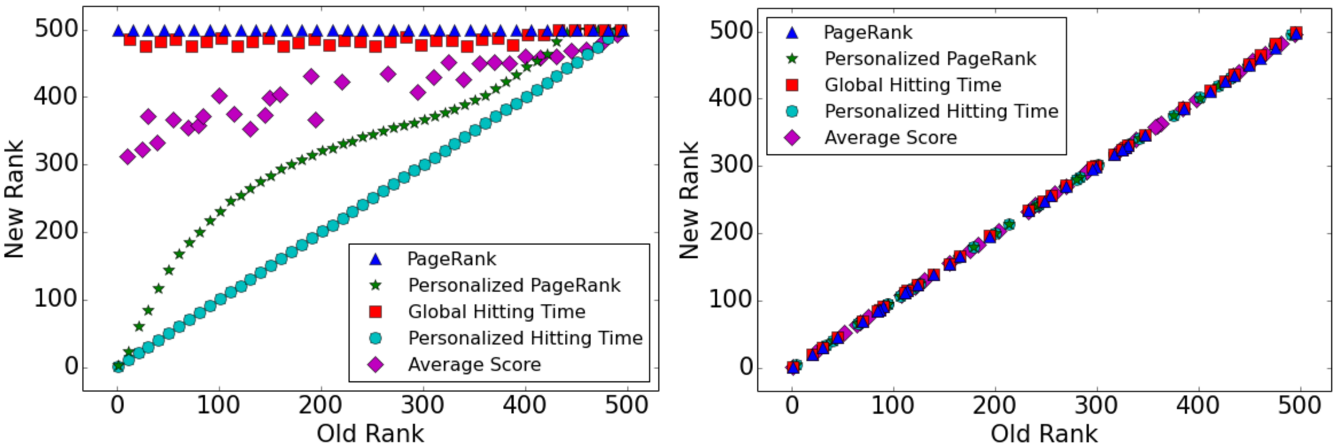
The important consequence of this result is that this random-sample based approximation approach can be used to scale the computation while obtaining the same strategic properties of the mechanism.

## EMPIRICAL ANALYSIS

In this section, we present the results of a quantitative study of the effect of manipulation in each mechanism, both in terms of the benefit it brings to an agent and the effect it has on informativeness.

[8] We scale the total number of walks in multi-walk as $20N^2$, which keeps the Spearman correlation between approximate and exact trust ranks between 0.99 and 0.995. The Spearman correlation between the rank order induced by approximate scores $(\hat{x}_{\text{PHT},i1}, \ldots, \hat{x}_{\text{PHT},iN})$ with exact scores $(x_{\text{PHT},i1}, \ldots, x_{\text{PHT},iN})$ is averaged over all agents.

[9] The exact and Monte Carlo algorithms were both implemented in Python for a fair comparison; there was no dependency on lower-level C or FORTRAN libraries. In practice (and in our other experiments), FORTRAN linear algebra routines can be used to substantially speed up the exact algorithm.



(a) with 50 Sybils

(b) without Sybils

**Figure 2: The effect of strategic behavior by a single agent, with $n = 500$ non-sybil agents, leading to graphs with up to $N = 550$ agents in total. (a) New rank vs old rank, with $50$ sybils. (b) New rank vs old rank, with $0$ sybils.**

## Experimental Set-up

Throughout, we compute exact trust scores and set $\alpha = 0.15$ for the random-walk based mechanisms, as is standard in the literature [9]. We let $\theta_i \sim uniform[0, 1]$.[10] The process of generating a trust graph captures the idea of transitivity: the higher the type of an agent $v_i$, the more likely $v_i$ has good information about other agents (i.e., the more likely $v_i$ actually trusts other trustworthy agents), and the more likely other agents are to trust $v_i$.

Concretely, this process is parameterized by integers $d > 0$ (the "out-neighborhood size") and $t > 0$ (the "transaction count"). The set $V_i$ of agents (size $|V_i| = d$) with whom agent $v_i$ initiates transactions is determined as follows. Until $d$ agents have been selected: with probability $\theta_i$, select some agent $v_j \in V$ according to soft-max probability $p_j = \exp(\theta_j/z)/\sum_{j'} \exp(\theta_{j'}/z)$ (we use $z = 0.05$); with probability $1 - \theta_i$, select some agent $v_j \in V$ uniformly at random.

Eventually, agent $v_i$ initiates a transaction with each $v_j \in V_i$. The belief that agent $v_i$ forms about the type of agent $v_j$ is defined as the average of $t$ independent Bernoulli trials, each succeeding with probability $\theta_j$ (we use $t = 8$, unless otherwise stated). This is then used as the edge weight $\beta_{ij}$ from $v_i$ to $v_j$ in the resulting trust graph.

Across our experiments, we vary the out-neighborhood size (i.e., $d$), the fraction of strategic agents, and the number of sybils available to each strategic agent.[11] Throughout, we let $n$ denote the non-sybil agents (strategic and non-strategic), which we vary between $n = 100$ and $n = 500$. We let $N$ denote the total number of agents in the graph (including the sybil agents created by the strategic agents), which varies between $N = 100$ and $N = 1100$.

## Effect of Manipulation

We first investigate the ability for an agent to change its rank in each mechanism. For this, we fix $n = 500$ and $d = 50$, so that each agent has transacted with 1/10 of the agents, and vary the number of sybils available to agents.[12]

For a fixed trust graph, we compute the effect on each agent's rank as if it were the only strategic agent and employed the optimal manipulation in the presence of sybils (following Theorem 1). If a strategic agent does not have any sybils, then it only drops all trust reports (or reports zero trust about others, in the case of AS).[13] For a personalized mechanism, we define the rank of an agent to be the average rank in all of its personalized rankings (excluding its own).

Figures 2(a) and (b) show the effect of manipulation on rank in each mechanism (500 is best, 1 is worst), where we have subsampled the results for better readability. Strategic agents have access to 50 sybils in (a), but we exclude sybils in (b). The results are averaged over three trials; e.g., the agent with "old rank" 20 has a rank of 20 without manipulation, and its "new rank" is the average of the ranks attained through manipulation in each of three trust graphs. We observe the following:

- From Figure 2(a) we observe that PHT is robust to manipulation, even in the presence of sybils, while the global mechanisms are more vulnerable to sybils. PPR is more robust than the global mechanisms, but more manipulable than PHT.
- From Figure 2(b) we observe that the mechanisms are all very robust to manipulation when agents cannot use sybils (i.e., when they can only cut outlinks). Note that cutting outlinks can only harm other agents' trust scores, and cannot boost one's own trust score. In reasonably well-connected graphs (which characterize our generated graphs), the capacity to harm others' trust scores in random walk-based mechanisms is limited, because the removal of one agent's edges has a negligible effect on the reachability of other agents.

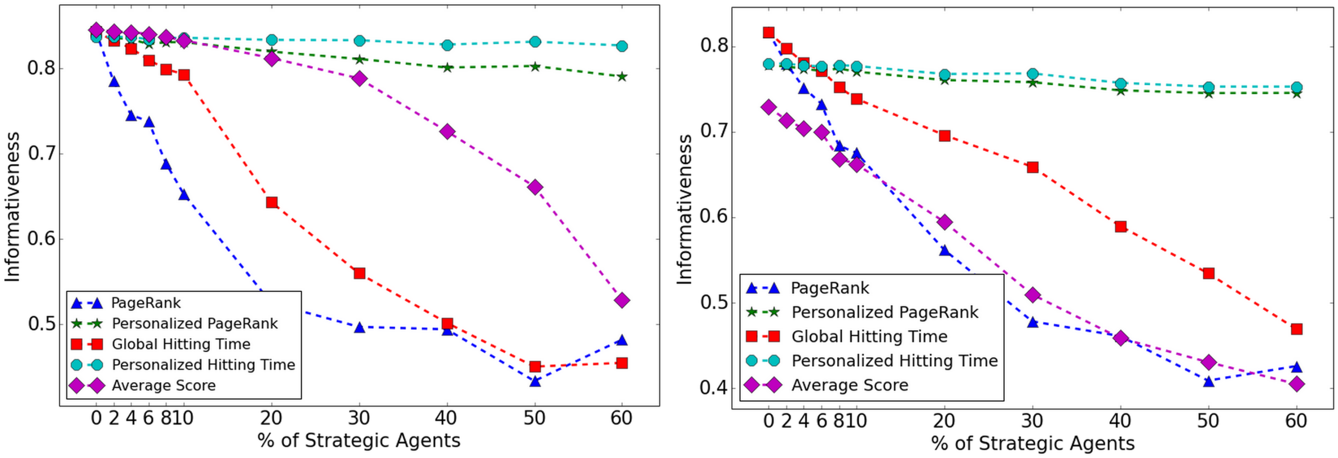
As we can see, the mechanisms differ primarily in their ability to handle sybils. The restart-capture effect in the global mechanisms is very powerful. The two-loop effect in PPR is also powerful relative to just dropping reports (compare Figure 2(a) and Figure 2(b)).

[10] We have also experimented with truncated Normal(0.5, 0.5), and Beta(2, 2), and they provide qualitatively similar results.

[11] We assume that the majority of the agents are non-strategic. This recognizes that some participants in real-world systems may be altruistic, but more importantly, that strategic behavior may be costly, and that strategic behavior could be identified and punished.

[12] The results are qualitatively the same for $n = 100$, $d = 5, d = 20, t = 2$ and $t = 32$.

[13] This is the optimal, no-sybil manipulation for all mechanisms except PageRank and PPR, where it may sometimes be useful to retain a single report. Because of this, the analysis of PageRank and PPR mechanisms in the special case of no sybils provides a lower-bound on the effect of optimal manipulation.



(a) Dense Graphs

(b) Sparse Graphs

**Figure 4: The effect of strategic behavior on the informativeness of each mechanism, with $n = 100$ non-sybil agents, leading to graphs with up to $N = 700$ agents in total. (a) Informativeness vs. fraction of strategic agents in dense graphs ($d = 30$) with $10$ sybils per strategic agent. (b) Informativeness vs. fraction of strategic agents in sparse graphs ($d = 5$) with $10$ sybils per strategic agent.**

In Figure 3, we vary the number of sybils available to the strategic agent and plot the increase in rank through strategic behavior, averaged across five strategic agents evenly distributed in initial ranking (initial ranks are 0, 100, 200, 300, and 400). Thus, the best possible average increase in rank is 300, which would occur if every agent achieved a new rank of 500. We see that sybils already have a large effect in all mechanisms except PHT even when an agent can only use a single sybil. The global mechanisms consistently perform worse than the personalized mechanisms because of restart-capture, and PageRank and PPR perform worse than the corresponding hitting time-based mechanisms because of two-loops. A strategic agent in PPR only requires a single sybil, and hence the two-loop manipulation explains well the difference between PPR and PHT, and also between PageRank and GHT (notice the similar differences between the corresponding curves in Figure 3).[14]

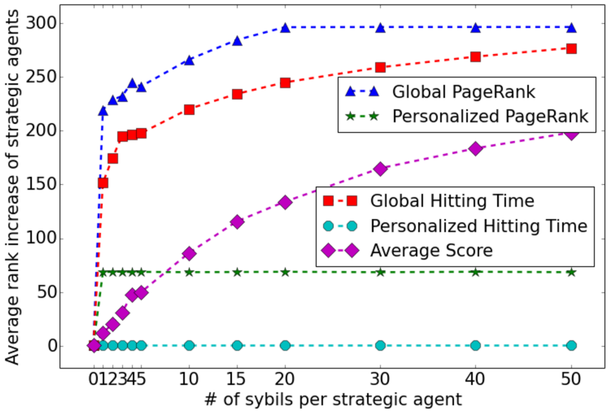


**Figure 3: The effect of strategic behavior with $n = 500$ non-sybil agents, leading to graphs with up to $N = 750$ agents in total. Shown is the average increase in rank vs. number of sybils available (averaged across 5 strategic agents).**

## Effect on Informativeness

Informativeness measures the effectiveness of a mechanism in allowing an agent to discriminate between high type and low type agents when initiating transactions. To measure this, we follow earlier work by Tang et al. [18]: We determine the fraction of transactions initiated by non-strategic agents that succeed, assuming that the counter-party is chosen as the agent with the highest trust score from a set of candidate agents, where this set comprises 5% of all agents selected uniformly at random. Our results are robust to the size of this set of candidate agents with whom the agents transact.

We focus on non-strategic agents because strategic agents receive no useful information from the personalized mechanisms, since the random walk initiated at such an agent cannot follow any edges. Furthermore, we are mainly interested in how cooperative agents are affected by manipulation by strategic agents.

For the following experiment, individual agents are designated to be *strategic* with probability $1 - \theta_i$ (such that agents with lower type are more likely to be strategic), capturing the intuition that lower type agents have more to gain from manipulating. However, the sampling is repeated until the desired number of strategic agents is achieved. The reported results are averaged over three trials.

We first consider the effect of varying the number of strategic agents, each of whom has access to 10 sybils. We use $n = 100$ non-sybil agents here, as the presence of the large number of strategic agents with 10 sybils each will lead to graphs with a maximum number of $N = 700$ total agents in Figure 4(a) and (b), and $N = 1100$ total agents in Figure 5.

We consider dense ($d = 30$) and sparse ($d = 5$) trust graphs, in Figure 4(a) and (b) respectively. The effect of graph density is to vary the amount of information available. We observe the following:

- PHT is the most informative mechanism when more than 4% and 10% of agents are strategic, in sparse and dense trust graphs respectively. With a very small number of strategic agents we

[14] This does not imply that restart-capture is “stronger” than two-loops. The effectiveness of restart-capture depends on the proportion of nodes that are sybils, while the effectiveness of a two-loop depends on the “original” reputation of the node, since two-loops have a “multiplicative” effect. Whether one is more effective than the other depends on the parameters of the particular graph.

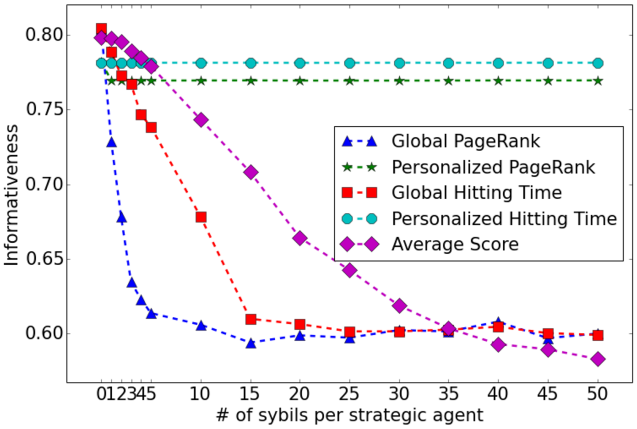


**Figure 5: The effect of strategic behavior on the informativeness of each mechanism, in systems with $n = 100$ non-sybil agents, leading to graphs with up to $N = 1100$ agents in total. Shown is the informativeness vs. number of sybils per strategic agent ($d = 20$ and 20 strategic agents).**

see that PageRank and GHT have a slight advantage for sparse graphs, with AS best-performing for dense graphs.[15]

- A large gap in performance opens up between the personalized trust mechanisms and the global mechanisms, even with as few as 10% of strategic agents.
- PHT is more informative than PPR, with its advantage getting larger as graphs become more dense and as the number of strategic agents increases.

Figure 5 shows how the informativeness changes when the number of sybils per strategic agent is varied, here using $d = 20$ and with 20% of the agents being strategic. PHT is better than the other random-walk mechanisms for just two sybils per agent. The AS mechanism's performance again dominates that of the global, random-walk based mechanisms for this relatively dense trust graph. The PHT mechanism has better informativeness than AS with four or more sybils per strategic agent.

In summary, Figures 4(a), 4(b), and Figure 5 clearly show the dominance of PHT over the other mechanisms in terms of informativeness. However, note that a comparison at any fixed percentage of strategic agents must be considered a lower bound on the difference we would expect to observe in practice. As the analysis in the previous section on "manipulation" has shown, under all mechanisms except PHT, the agents have a large incentive to become strategic. Consider again Figure 3, where a single sybil per strategic agent is enough to increase the rank of a strategic agent under PPR by roughly 70. Thus, in equilibrium, we should expect a large number of strategic agents when using GHT, PR, PPR, while we should expect no strategic agents when using PHT. Thus, for the ultimate comparison of informativeness, we must compare the performance of PHT in Figures 4(a) and (b) with 0% strategic agents with the performance of the other mechanisms with a larger percentage of strategic agents. This elevates the difference in informativeness between PHT and the other mechanisms even further.

# CONCLUSION

In this paper, we have presented the first study of the *personalized hitting time (PHT)* trust mechanism. We have made three main contributions. First, we have provided a Monte Carlo approximation algorithm to compute PHT scores efficiently, with good theoretical bounds on its approximation error. Second, we have shown formally that PHT is significantly more robust against sybil attacks than all other mechanisms we have studied. In particular, we haven proven that the optimal manipulation under PHT only involves dropping outlinks, but does not involve adding sybils. Furthermore, we have shown that PHT retains this robustness to sybil attacks when PHT scores are approximated. Third, and most importantly, we have provided an empirical evaluation, showing the impact that strategic agents (that can create sybils) have on PHT, global PageRank, personalized PageRank, and global hitting time (GHT). Our experimental results are striking in demonstrating the devastating effect that sybils have on the manipulability and the informativeness of existing mechanisms, even with just a few sybils per agent. This is particularly true for the global mechanisms, including GHT, whose informativeness rapidly declines as the number of strategic agents, or the number of sybils per strategic agents grows. In contrast, because adding sybils is not beneficial under PHT, it remains highly informative, even in the presence of a large number of strategic agents. We found this large dominance of PHT over GHT particularly surprising, given the small difference between the two mechanisms. The high manipulability and low informativeness of GHT raises doubts about the applicability of the GHT mechanism in practice, and leads us to suggest PHT as a building block for multi-agent systems. Future research should experiment on real-world networks (e.g., leveraging some network data at KONECT [11] or SNAP [13]), validating the robustness, informativeness and computational properties of PHT at a larger scale.

# Acknowledgements

We are very grateful for the feedback we have received from reviewers on previous versions of this paper.

# REFERENCES

[1] A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, Oct. 1999.

[2] M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM Transactions on Internet Technology*, 5(1):92–128, 2005.

[3] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems*, pages 128–132. ACM, 2005.

[4] A. Cheng and E. Friedman. Manipulability of PageRank under sybil strategies. In *First Workshop on the Economics of Networked Systems*, 2006.

[5] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-Law Distributions in Empirical Data. *SIAM Review*, 51(4):661–703, Nov. 2009.

[6] J. W. Demmel, M. T. Heath, and H. A. van der Vorst. Parallel numerical linear algebra. *Acta Numerica*, 2:111–197, 1993.

[7] D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós. Towards scaling fully personalized PageRank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2(3):333–358, 2005.

[15] Two factors are at play. First, the locality of AS (just averaging local information) means that the effect of strategic behavior is isolated (just dropping reports from strategic agents and adding fake reports to these same agents). Second, AS is effective in dense graphs because the average score is accurate, consisting of an average over 8 × 30 unbiased samples ($t = 8$ and $d = 30$).

[8] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating Web Spam with TrustRank . In *Proceedings of the 30th International Conference on Very Large Databases*, pages 576–587, Mar. 2004.

[9] J. Hopcroft and D. Sheldon. Manipulation-resistant reputations using hitting time. In *Algorithms and Models for the Web-Graph*, pages 68–81, 2007.

[10] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International Conference on World Wide Web*, pages 640–651, 2003.

[11] J. Kunegis. The koblenz network collection (KONECT), 2015. Accessed on August 30, 2015, `http://konect.uni-koblenz.de/`.

[12] R. Landa, D. Griffin, R. G. Clegg, E. Mykoniati, and M. Rio. A sybilproof indirect reciprocity mechanism for peer-to-peer networks. In *INFOCOM 2009*, pages 343–351, 2009.

[13] J. Leskovec. Stanford network analysis project, 2015. Accessed on August 30, 2015, `http://snap.stanford.edu/index.html`.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[15] J. S. Rosenschein and G. Zlotkin. Designing conventions for automated negotiation. *AI Magazine*, 15(3):29, 1994.

[16] S. Seuken and D. C. Parkes. Sybil-proof accounting mechanisms with transitive trust. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Paris, France, 2014.

[17] S. Seuken, J. Tang, and D. C. Parkes. Accounting mechanisms for distributed work systems. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, 2010.

[18] J. Tang, S. Seuken, and D. C. Parkes. Hybrid transitive trust mechanisms. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 233–240, 2010.

[19] M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false-name bids in combinatorial auctions: new fraud in internet auctions. *Games and Economic Behavior*, 46(1):174–188, 2004.

[20] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. *Security and Privacy*, pages 3–17, 2008.

[21] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: Defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36(4):267–278, 2006.

[22] H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser. A survey of multi-agent trust management systems. *IEEE Access*, pages 35–50, 2013.