# Hybrid Transitive Trust Mechanisms

Jie Tang
Computer Science Division
University of California,
Berkeley
jietang@eecs.berkeley.edu

Sven Seuken
School of Engineering
& Applied Sciences
Harvard University
seuken@eecs.harvard.edu

David C. Parkes
School of Engineering
& Applied Sciences
Harvard University
parkes@eecs.harvard.edu

## ABSTRACT

Establishing trust amongst agents is of central importance to the development of well-functioning multi-agent systems. For example, the anonymity of transactions on the Internet can lead to inefficiencies; e.g., a seller on eBay failing to ship a good as promised, or a user free-riding on a file-sharing network. Trust (or reputation) mechanisms can help by aggregating and sharing trust information between agents. Unfortunately these mechanisms can often be manipulated by strategic agents. Existing mechanisms are either very robust to manipulation (i.e., manipulations are not beneficial for strategic agents), or they are very informative (i.e., good at aggregating trust data), but never both. This paper explores this trade-off between these competing desiderata. First, we introduce a metric to evaluate the informativeness of existing trust mechanisms. We then show analytically that trust mechanisms can be combined to generate new *hybrid mechanisms* with intermediate robustness properties. We establish through simulation that hybrid mechanisms can achieve higher overall efficiency in environments with risky transactions and mixtures of agent types (some cooperative, some malicious, and some strategic) than any previously known mechanism.

## Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

## General Terms

Algorithms, Design, Economics

## Keywords

Trust, Reputation, Mechanism Design, Informativeness

## 1. INTRODUCTION

We often interact with anonymous parties over the Internet and in many environments this can lead to fraudulent behavior. For example, on e-commerce websites a seller might advertise a product with false information, or in P2P networks a malicious user might distribute a virus. Online, it is difficult to know whom to trust. Information from other users with previous experience in the same online system can help separate malicious from trustworthy users and incentivize all users to act cooperatively. On eBay for example, user feedback about the quality of sellers and buyers is aggregated. Research has shown that consumers take the aggregated information regarding a seller into account when purchasing products [12]. Mechanisms that aggregate information and compute a score for each agent are called *trust mechanisms* (or *reputation mechanisms*).[1] In this paper we focus on the design of *transitive trust* mechanisms, i.e., we assume that if agent A trusts B, and B trusts C, then A also trusts C to some degree.

### 1.1 Informativeness vs. Strategyproofness

We aim to design trust mechanisms that have good *informativeness* as well as *strategyproofness* properties. A mechanism is informative if it aggregates the available information well, such that agents using it can successfully separate good from bad trading partners. A mechanism is strategyproof if agents cannot improve their utility in the system by manipulating the trust mechanism. Strategyproofness is important here because we consider mechanisms that must rely on information provided voluntarily by the agents and where the outcome of individual transactions cannot be monitored centrally. Depending on the particular trust mechanism, agents might be able to manipulate by spreading bad information about other agents in the system, or by creating fake agents (sybils) that spread good information about themselves.

Existing trust mechanisms represent distinct tradeoffs between robustness and informativeness. This can be problematic for overall system efficiency. On the one hand, if a mechanism is not informative then it is not very helpful in identifying good and bad agents, resulting in poor trading decisions and low overall efficiency. On the other hand, if a mechanism can be easily manipulated, then many agents may choose to influence a mechanism to their advantage, which in turn decreases overall efficiency as well. In real environments with risky transactions, there is likely to be a mixture of different kinds of agents. Some agents will be highly trustworthy and *cooperative*, likely to complete a transaction in good faith. Some agents will be less trustworthy and *malicious*, with a greater probability of participating in an incomplete or fraudulent transaction. Depending on how costly manipulations are, some of the malicious agents will act *strategically* and manipulate a trust mechanism to their advantage.

Previous research has primarily focused on a formal analysis of the strategyproofness properties of different mechanisms. However, a formal instrument for measuring and comparing informativeness was missing. In this paper, we propose a simple metric for measuring the informativeness of a trust mechanism, independent from how this information is being used for making decisions in the environment. This gives us a way to evaluate how well different

[1] The terminology is in fact used more or less interchangeably in the literature. Here we use “trust mechanisms” because we use the concept of transitive trust.

**Cite as:** Hybrid Transitive Trust Mechanisms, Jie Tang, Sven Seuken, David C. Parkes, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 233-240

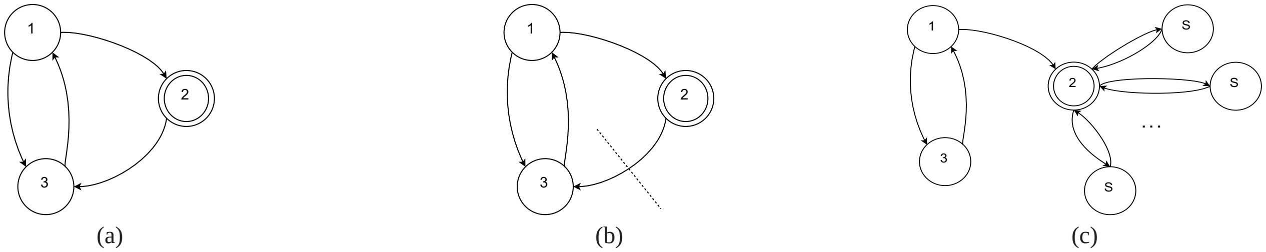


**Figure 1: (a) A simple trust graph with three agents (edge weights are omitted). (b) Agent 2 manipulated the trust graph by cutting its outlink to agent 3, i.e., decreasing its trust report to 0. (c) Agent 2 manipulated the trust graph further by adding sybil agents.**

mechanisms aggregate trust information. We then combine existing transitive trust mechanisms introducing new *hybrid* transitive trust mechanisms, enabling a new continuum of tradeoffs between the competing desiderata of informativeness and strategyproofness. This is desirable in order to make the tradeoff that is best for a given environment with a particular agent population. We establish analytically that these hybrid mechanisms have intermediate strategyproofness properties and we show experimentally that they also have good informativeness properties. Ultimately, however, we are interested in the overall efficiency resulting from the use of hybrid mechanisms. We study this in two different simulated domains (file-sharing with viruses, and website surfing). Our results show that in some settings, hybrid mechanisms can outperform previously known mechanisms, with efficiency gains up to 7%.

## 1.2 Related Work

Many transitive trust mechanisms have been introduced in the literature (for a recent survey see Friedman et al. [8]). The most well known mechanism is PageRank [10] originally used by Google to rank websites. However, PageRank was soon found to be highly susceptible to manipulation, and thus subsequent work has primarily focused on solving the manipulability problem [6, 7, 13]. Altman et al. [1] presented the first axiomatic approach to the design of trust mechanisms, providing systematic insight into the design space. Guha et al. [9] present the first large-scale empirical study on trust mechanisms using transitive trust networks. Sami and Resnick [11] study the dynamics of transitive trust mechanisms in environments with risky transactions, looking to limit the cumulative effect of an attack by a powerful adversary.

# 2. TRANSITIVE TRUST MECHANISMS

We consider multi-agent systems where agents engage in risky transactions with many other agents, but rarely have repeat interactions with the same other agent. An agent who contacts another agent puts itself at risk in terms of whether the second agent will complete the transaction correctly or not. A good outcome leads to a gain in utility by the first agent, a bad outcome to a loss in utility.

DEFINITION 1 (AGENT MODEL). *Each agent $v_i$ has a (private) type $\theta_i \in [0,1]$, which represents its goodness, or trustworthiness. This is the probability that an agent will generate a good outcome when participating in a transaction with another agent.*

By sharing their direct experiences via the trust mechanism, the agents can help each other identify and thus avoid bad agents.

DEFINITION 2 (AGENT INFORMATION & REPORTS). *Given a set of agents $V = \{v_1, ..., v_n\}$, let $V_i$ denote the agents that $v_i$ has direct trust information $t_i$ about, where $t_i : V_i \to [0,1]$, i.e., $t_i(v_j)$ is the trust agent $v_i$ has in agent $v_j$. Agent $v_i$ makes reports $(\hat{V}_i, \hat{t}_i)$ to a transitive trust mechanism. Agent $v_i$ is truthful if and only if $(V_i, t_i) = (\hat{V}_i, \hat{t}_i)$.*

DEFINITION 3 (TRUST GRAPH). *A trust graph $G = (V, E, w)$ is a set of vertices $V$ and directed edges $(v_i, v_j) \in E, v_i, v_j \in V$. Each edge $(v_i, v_j)$ has an associated weight $w(v_i, v_j) \in [0,1]$.*

In a trust graph, vertices are individual agents, and the weight of an edge $(v_i, v_j)$ corresponds to the last claim that $v_i$ has made regarding its direct trust in agent $v_j$ (see Figure 1(a) for a simple example). To simplify notation we sometimes use $i, j$ directly instead of $v_i, v_j$. A trust graph is constructed to correspond to agent reports as follows: for each vertex $v_i$, given report $(\hat{V}_i, \hat{t}_i)$, create a directed edge $(v_i, v_j) \in E$ for each $v_j \in \hat{V}_i$ and define $w(v_i, v_j) = \hat{t}_i(v_j)$. If agent $v_i$ has reported truthfully, we call the corresponding trust graph a *$v_i$-truthful trust graph*. If all agents have reported truthfully, we call the corresponding trust graph a *truthful trust graph*.

DEFINITION 4 (TRANSITIVE TRUST MECHANISM). *Let $\mathcal{G}_\mathcal{V}$ denote the set of trust graphs $G = (V, E, w)$ on $V$. A transitive trust mechanism $M$ is a function that for every set of agents $V$ and for every individual agent $v_i \in V$ maps $\mathcal{G}_\mathcal{V}$ to a vector of trust scores for all other agents $v_j \in V, v_j \neq v_i$. More formally: $M : \mathcal{G}_\mathcal{V} \times V \to [0,1]^{n-1}$. Each $M_j(G, v_i)$ denotes the trust score assigned to agent $v_j$ from the perspective of $v_i$. We let $M(G, v_i)$ denote the vector of all trust scores from agent $v_i$'s perspective.*

This allows for personalized trust mechanisms where the trust score assigned to some agent $v_j$ depends on which agent's perspective $v_i \neq v_j$ is adopted. This might make sense for an environment where agents trust their own direct experiences more than the reported experiences of others

The goal of using a trust mechanism is to maximize overall system efficiency. We measure the efficiency of a trust mechanism as the fraction of transactions by non-strategic agents that are successful. This depends on the strategyproofness and informativeness properties of the mechanism as well as the details of each problem domain. The strategyproofness and informativeness of a mechanism are formally defined in Sections 2.1 and 4, respectively.

## 2.1 Manipulations and Strategyproofness

Following earlier work, we consider two different classes of manipulations by strategic agents.

DEFINITION 5 (MISREPORT). *Given trust graph $G = (V, E, w)$, define the set $E_{-v} = \{(x, y) : (x, y) \in E, x \neq v\}$ (i.e., the set of all edges in G that do not start at v). A misreport strategy for agent $v \in V$ is a tuple $\sigma = (E_v, w_v)$ where $E_v = \{(v, u) : u \in V\}$ and $w_v : E_v \to [0,1]$. Applying the strategy $\sigma$ to $G$ results in trust graph $G \downarrow \sigma = G' = (V, E_{-v} \cup E_v, w')$ where $w'(e) = w(e)$ for all $e \in E_{-v}$, and $w'(e') = w_v(e')$ for all $e' \in E_v$.*

See Figure 1(b) for an example of a misreport attack.

We now define a sybil manipulation (see Cheng and Friedman [6]) which involves the creation of multiple fake nodes and associated fake edges in the trust graph. Figure 1(c) shows an example of a sybil manipulation.

DEFINITION 6 (SYBIL MANIPULATION). *Given a trust graph $G = (V, E, w)$, a sybil manipulation for agent $v \in V$ is a tuple $\sigma = (S, E_S, w_S)$ where $S = \{s_1, ..., s_m\}$ is a set of sybil agents, $E_S$ is a set of edges $E_S = \{(x, y) : x \in S \cup \{v\}, y \in V \cup S\}$, and $w_S : E_S \rightarrow [0, 1]$ are the weights on the edges in $E_S$. Applying the sybil manipulation $\sigma$ to $G$ results in a modified trust graph $G \downarrow \sigma = G' = (V \cup S, E \cup E_S, w')$, where $w'(e) = w(e)$ for $e \in E$, and $w'(e') = w_S(e')$ for $e' \in E_S$.*

Note that in general, an agent can manipulate a trust mechanism via a combination of misreports and sybil manipulations. For these combinations, $G \downarrow \sigma$ is defined analogously.

We can now define appropriate concepts of strategyproofness. We use two different concepts, similar to the ones introduced in Cheng and Friedman [6]. The first one, rank-strategyproofness, compares the relative trust scores of agents. The second one, value-strategyproofness, considers an agent's absolute trust score.

DEFINITION 7 (RANK-STRATEGYPROOF). *A transitive trust mechanism is rank-strategyproof if for any $v_i$-truthful trust graph $G = (V, E, w)$ where $v_i \in V$, and for every strategy $\sigma$ by node $v_i$ s.t. $G \downarrow \sigma = G'$, for all $v_j \neq v_i$, for all $v_k \neq v_i$ : $M_i(G, v_j) < M_k(G, v_j) \Rightarrow M_i(G', v_j) < M_k(G', v_j)$, i.e., an agent cannot increase its position in a rank-order from the perspective of any such agent $v_j \neq v_i$.*

DEFINITION 8 (VALUE-STRATEGYPROOF). *A transitive trust mechanism is value-strategyproof if for any $v_i$-truthful trust graph $G = (V, E, w)$ with $v_i \in V$, and for every strategy $\sigma$ by node $v_i$ s.t. $G \downarrow \sigma = G'$, for all $v_j \neq v_i$: $M_i(G, v_j) \geq M_i(G', v_j)$, i.e., an agent cannot increase its absolute trust score from the perspective of any agent $v_j \neq v_i$.*

Rank-strategyproofness is appropriate, for example, when an agent can choose from a list of agents and only the *relative* trust scores are important to identify the most trustworthy one. Value-strategyproofness is appropriate, for example, when agents use a threshold approach to decide which other agents to transact with; e.g., any agent with a trust score above a threshold may be acceptable. It is easy to show that neither of these concepts dominates one another. For many applications, however, rank-strategyproofness is a more natural requirement, but it is also harder to achieve.

## 2.2 Existing Transitive Trust Mechanisms

We now review four transitive trust mechanisms that have been introduced in this form or very similarly before. The trust scores produced by the mechanisms are normalized to be in $[0, 1]$.

DEFINITION 9 (PAGERANK [10]). *Given a trust graph $G = (V, E, w)$, PageRank conducts a random walk from a random node $v_i \in V$ that at each step, with probability $\lambda$ (for $\lambda \in [0, 1)$) follows a random outlink with probability proportional to weight $w(v_i, v_j)$, as a fraction of the total weight on all outlinks, and with probability $1 - \lambda$ jumps to another node with uniform probability. If the random walk reaches a node with no outgoing links then PageRank randomly jumps to another node in the trust graph with uniform probability. The trust score $M_j(G, v_i) = \pi(G, v_j)$ of a node $v_j$ is the same, irrespective of $v_i$, and is given by the probability $\pi(G, v_j)$ of being in node $v_j$ in the stationary distribution of the Markov process described by the random walk.*

Some mechanisms use pre-trusted nodes in their algorithms. This is reasonable for many domains, e.g., in P2P networks the administrator of the mechanism might own some trusted servers.

DEFINITION 10 (HITTINGTIME [13]). *Given a trust graph $G$, the hitting time of a node $v_j$, $H(v_j)$, is the number of steps before a random walk on $G$ first reaches $v_j$. A hitting time trust mechanism has a set of pre-trusted nodes, and after each time step, the random walk jumps back to one of the pre-trusted nodes with some probability $\lambda$. The random variable $J$ denotes the number of time steps before the random walk performs a jump. The trust score of node $v_j$ is the probability that the random walk reaches $v$ before jumping, i.e., $\forall i : M_j(G, v_i) = Pr(H(v_j) < J)$.*

DEFINITION 11 (MAXFLOW MECHANISM [6]). *Given a trust graph $G = (V, E, w)$ and nodes $v_i, v_j \in V$, let $MF(v_i, v_j)$ denote the maximum flow from node $v_i$ to node $v_j$. The maxflow transitive trust mechanism sets $M_j(G, v_i) = MF(v_i, v_j)$.*

DEFINITION 12 (SHORTESTPATH MECHANISM [3]). *Given a trust graph $G = (V, E, w)$, define the trust graph $G' = (V, E, w')$ with $w'(i, j) = \frac{1}{w(i,j)}$, i.e., all edge weights are flipped such that low trust scores lead to high edge weights in $G'$. Now, let $SP_{G'}(v_i, v_j)$ denote the length of the shortest path between agents $v_i$ and $v_j$ in $G'$. The shortest-path mechanism sets $M_j(G, v_i) = \frac{1}{SP_{G'}(v_i, v_j)}$.*

Each of theses mechanisms makes a distinct tradeoff between informativeness and strategyproofness. Previous research has already established their strategyproofness properties: ShortestPath is best being rank-strategyproof and value-strategyproof; MaxFlow and HittingTime are both value-strategyproof, and finally PageRank is last with no formal strategyproofness properties (see Table 1).

| Mechanism | Rank-SP | Value-SP |
|---|---|---|
| ShortestPath | Yes | Yes |
| MaxFlow | No | Yes |
| HittingTime | No | Yes |
| PageRank | No | No |

**Table 1: Strategyproofness of Existing Trust Mechanisms**

We investigate the informativeness properties of all four mechanisms in Section 4. We find that the order of the mechanisms with respect to informativeness is roughly reversed. This makes intuitive sense: the more information a mechanism ignores when computing trust scores, the better its strategyproofness properties but the worse its informativeness properties. This illustrates the trade-off we make when designing trust mechanisms.

# 3. HYBRID MECHANISMS

We now introduce the idea of a hybrid transitive trust mechanism, which is defined as a linear combination of two mechanisms.

DEFINITION 13 (HYBRID TRANSITIVE TRUST MECHANISMS). *Given mechanisms $M^1$ and $M^2$, we let $M^\alpha(M^1, M^2)$ denote the $\alpha$-hybrid of those mechanisms. Given a trust graph $G = (V, E, w)$ and $v_i, v_j \in V$, let $M_j^1(G, v_i)$ denote the trust value of $v_j$ from $v_i$'s perspective under $M^1$, and let $M_j^2(G, v_i)$ denote the trust value of $v_j$ from $v_i$'s perspective under $M^2$. The reputation of $v_j$ from $v_i$'s perspective under $M^\alpha(M^1, M^2)$ is*

$$M_j^\alpha(G, v_i) = (1 - \alpha)M_j^1(G, v_i) + \alpha M_j^2(G, v_i).$$

For a hybrid mechanism $M_\alpha(M^1, M^2)$ we will by convention always combine two mechanisms in which $M^1$ is more strategyproof than $M^2$. Often times, but not always, $M^2$ will be more informative than $M^1$. Thus, as $\alpha$ is increased from 0 to 1, the opportunities for manipulation increase, but we also expect the mechanism to become more informative, at least when no strategic agents are present. We will look for non-trivial hybrids (with $0 < \alpha < 1$) that have better efficiency than either extreme mechanism.

### 3.1 Strategyproofness of Hybrid Mechanisms

LEMMA 1. *If mechanisms $M^1$ and $M^2$ are value-strategyproof, then $M^\alpha(M^1, M^2)$ is value-strategyproof.*

PROOF. If $M^1$ and $M^2$ are both value-strategyproof, then for any $v_i$-truthful trust graph $G = (V, E, w)$ with $v_i \in V$, for every strategy $\sigma$ by node $v_i$ s.t. $G \downarrow \sigma = G'$, for all $v_j \neq v_i$, we have $M_i^1(G, v_j) \geq M_i^1(G', v_j)$ and $M_i^2(G, v_j) \geq M_i^2(G', v_j)$. Thus, it follows that $(1-\alpha)M_i^1(G, v_j) + \alpha M_i^2(G, v_j) \geq (1-\alpha)M_i^1(G', v_j) + \alpha M_i^2(G', v_j)$, for any $\alpha \in [0, 1]$. □

Unfortunately this does not hold true for the property of rank-strategyproofness.

LEMMA 2. *If mechanisms $M^1$ and $M^2$ are rank-strategyproof, then $M^\alpha(M^1, M^2)$ is not necessarily rank-strategyproof.*

PROOF. By counterexample. Assume a truthful trust graph with two agents 1 and 2 and with only one edge from agent 1 to agent 2. $M^1$ always assigns a trust score of 1 to agent 2 and a trust score of 0.2 to agent 1 (and all other agents). $M^1$ is trivially rank-strategyproof. $M^2$ always assigns a trust score of 1 to agent 1, and assigns trust score 0.5 to agent 2 if an edge exists from agent 1 to agent 2 and trust score 0 otherwise. $M^2$ is rank-strategyproof because agent 1 is always the highest-ranked agent, and agent 2 cannot affect the final ranking. Now, for $\alpha = 0.5$, agent 1 has trust value $0.6$ while agent 2 has trust value $0.75$. If agent 1 now removes the link to agent 2, then agent 2's trust value is lowered to $0.5$, and agent 1 becomes ranked higher than agent 2, thus proving that $M^\alpha(M^1, M^2)$ is not rank-strategyproof. □

For the design of hybrid mechanisms, we adopt relaxed notions of strategyproofness (similar to concepts adopted by [2]).

DEFINITION 14 ($\varepsilon$-VALUE-STRATEGYPROOFNESS). *A tran-sitive-trust mechanism is $\varepsilon$-value-strategyproof for $\varepsilon > 0$ if for any $v_i$-truthful trust graph $G = (V, E, w)$ with $v_i \in V$ and for all manipulation strategies $\sigma$ for $v_i$ giving $G' = G \downarrow \sigma$, for all $v_j \neq v_i$, $M_i(G, v_j) + \varepsilon \geq M_i(G', v_j)$.*

DEFINITION 15 ($\varepsilon$-RANK-STRATEGYPROOFNESS). *A transitive-trust mechanism is $\varepsilon$-rank-strategyproof for $\varepsilon > 0$ if for any $v_i$-truthful trust graph $G = (V, E, w)$ with $v_i \in V$ and for all manipulation strategies $\sigma$ for $v_i$ s.t. $G' = G \downarrow \sigma$, for all $v_j \neq v_i, v_k \in V$, $M_i(G, v_j) + \varepsilon \leq M_k(G, v_j) \Rightarrow M_i(G', v_j) \leq M_k(G', v_j)$.*

In words, an $\varepsilon$-value-strategyproof mechanism is one in which an agent cannot increase its trust score by more than $\varepsilon$ under any manipulation strategy and for any trust graph. An $\varepsilon$-rank-strategyproof mechanism is one in which an agent cannot overcome more than a difference of $\varepsilon$ in trust scores between itself and any other agent, whatever the trust graph and for any manipulation strategy.

### 3.2 Value-Strategyproofness Results

THEOREM 1. *If transitive trust mechanisms $M^1$ and $M^2$ are $\varepsilon_1$ and $\varepsilon_2$-value-strategyproof respectively, then $M^\alpha(M^1, M^2)$ is $\big((1-\alpha)\varepsilon_1 + \alpha\varepsilon_2\big)$-value-strategyproof.*

PROOF. Let $M_i^1, M_i^2$ denote the trust scores of $v_i$ (as viewed by some other agent) under mechanisms $M^1$ and $M^2$ when $v_i$ is truthful. Let $M_i^\alpha = (1-\alpha)M_i^1 + \alpha M_i^2$. Let $\overline{M_i^\alpha}, \overline{M_i^1}$ and $\overline{M_i^2}$ denote the trust scores after $v_i$ has performed manipulations. Then:

$$\begin{aligned}
&\overline{M_i^\alpha} - M_i^\alpha = \\
&(1-\alpha)(\overline{M_i^1} - M_i^1) + \alpha(\overline{M_i^2} - M_i^2) \\
&\quad \leq (1-\alpha)\varepsilon_1 + \alpha\varepsilon_2,
\end{aligned}$$

and we see that $M^\alpha$ is $\big((1-\alpha)\varepsilon_1 + \alpha\varepsilon_2\big)$-value-strategyproof. □

We can now prove corollaries for specific hybrid trust mechanisms:

COROLLARY 1. *$M^\alpha$(Hitting, PageRank) is $0.5\alpha$-value-strategyproof.*

PROOF. The HittingTime mechanism is value-strategyproof [13]. Moreover, Bianchini et al. [4] establish that PageRank is $0.5$-value-strategyproof. By Theorem 1, we have that $M^\alpha$(Hitting, PageRank) is $0.5\alpha$-value-strategyproof. □

COROLLARY 2. *$M^\alpha$(MaxFlow, PageRank) is $0.5\alpha$-value-strategyproof.*

COROLLARY 3. *$M^\alpha$(Shortest, PageRank) is $0.5\alpha$-value-strategyproof.*

PROOF. MaxFlow and ShortestPath are both value-strategyproof and thus Corollaries 2 and 3 also follow from Theorem 1. □

### 3.3 Rank-Strategyproofness Results

Establishing rank-strategyproofness properties for hybrid transitive-trust mechanisms requires a more delicate argument. For this, we introduce the following property:

DEFINITION 16 (UPWARDS VALUE-PRESERVANCE). *A tran-sitive-trust mechanism is upwards value-preserving if for any trust graph $G = (V, E, w)$, for any $v_i \in V$, for every strategy $\sigma$ by node $v_i$ s.t. $G \downarrow \sigma = G'$, for all $v_j \neq v_i$, for all $v_k \neq v_i$ we have $M_k(G, v_j) > M_i(G, v_j) \Rightarrow M_k(G', v_j) \geq M_k(G, v_j)$.*

This property requires that an agent cannot decrease the trust score of a higher ranked agent. Note that the ShortestPath mechanism is easily seen to be upwards value-preserving: if $v_i$ has a lower trust score than $v_k$ from $v_j$'s perspective, then the path from $v_j$ to $v_k$ is shorter than then path from $v_j$ to $v_i$; thus, $v_i$ cannot be on the path between agents $v_k$ and $v_j$, and therefore $v_i$ cannot affect $v_k$'s trust score.[2]

THEOREM 2. *If transitive trust mechanisms $M^1$ and $M^2$ are value-strategyproof and $M^1$ satisfies upwards value-preservance, then $M^\alpha(M^1, M^2)$ is $\alpha$-rank-strategyproof.*

PROOF. We analyze the trust scores of agents $v_i$ and $v_j$ from any third agent's perspective. To simplify notation, let $M_i^1, M_j^1$ denote the trust scores of $v_i, v_j$ under $M^1$ and let $M_i^2, M_j^2$ denote the trust scores under $M^2$. Let $M_i^\alpha, M_j^\alpha$ denote the trust scores under $M_\alpha$. Furthermore, let $\overline{M_i^1}, \overline{M_j^1}, \overline{M_i^2}, \overline{M_j^2}$, and $\overline{M_i^\alpha}, \overline{M_j^\alpha}$ denote the analogous trust scores after $v_i$ has performed manipulations. WLOG, assume that $M_i^\alpha > M_j^\alpha$, i.e., $(1-\alpha)M_i^1 + \alpha M_i^2 > (1-\alpha)M_j^1 + \alpha M_j^2$. With this assumption, it impossible that both $M_i^1 < M_j^1$ and $M_i^2 < M_j^2$. Thus, we only need to consider the following two cases:

**Case 1:** $M_i^1 > M_j^1$: Because $M^1$ and $M^2$ are both value-strategyproof, agent $v_j$ cannot increase its own trust score, i.e.,

[2] However, not all rank-strategyproof mechanisms are upwards value-preserving. Consider a simple example with 2 agents $v_1, v_2$. Consider the trust mechanism $M$ which assigns trust scores $0.1, 0.2$ to agents $v_1, v_2$, respectively, unless the only edge in the graph is the edge $(v_1, v_2)$ in which case $M$ assigns trust scores $0.2, 0.4$ to agents $v_1, v_2$. Note that $v_2$ always has a higher trust score than $v_1$, so this mechanism is rank-strategyproof. However, it is not upwards value-preserving: if we start out with a graph with the single edge $(v_1, v_2)$, then $v_1$ can decrease the trust score of $v_2$ from $0.4$ to $0.2$ by cutting its outlink $(v_1, v_2)$.

$\overline{M_j^\alpha} \leq M_j^\alpha$. Because $M^1$ is upwards value-preserving, agent $v_j$ also cannot decrease $v_i$'s trust score under $M^1$. However, agent $v_j$ can decrease agent $v_i$'s trust score under $M^2$. But $M_i^2 - \overline{M_i^2} \leq 1$ since $M_i^2 \leq 1$ and $\overline{M_i^2} \geq 0$. So, we have that $M_i^\alpha - \overline{M_i^\alpha} \leq \alpha$. Putting all these arguments together we get:$\overline{M_i^\alpha} - \overline{M_j^\alpha} \geq \overline{M_i^\alpha} - M_j^\alpha \geq M_i^\alpha - \alpha - M_j^\alpha \geq \alpha - \alpha = 0$. And thus, $M^\alpha(M^1, M^2)$ is $\alpha$-rank-strategyproof in case 1.

**Case 2:** $M_i^1 < M_j^1$ and $M_i^2 > M_j^2$: For $\alpha = 0$ or $\alpha = 1$ there is nothing to be shown. For $0 < \alpha < 1$ we show that $M_i^\alpha - M_j^\alpha \geq \alpha$ is impossible to begin with:

$$\begin{aligned}
&M_i^\alpha - M_j^\alpha \\
&= \alpha M_i^2 + (1-\alpha)M_i^1 - \alpha M_j^2 - (1-\alpha)M_j^1 \\
&\leq \alpha + (1-\alpha)M_i^1 - (1-\alpha)M_j^1 \\
&= \alpha - (1-\alpha)(M_j^1 - M_i^1) < \alpha.
\end{aligned}$$

Thus, $M^\alpha$ is $\alpha$-rank-strategyproof in case 2 as well. □

COROLLARY 4. *Hybrid mechanism $M^\alpha$(Shortest, MaxFlow) is $\alpha$-rank-strategyproof.*

COROLLARY 5. *Hybrid mechanism $M^\alpha$(Shortest, Hitting) is $\alpha$-rank-strategyproof.*

PROOF. ShortestPath, MaxFlow, and Hitting-Time are value-strategyproof [3, 7, 13]. Moreover, ShortestPath is upwards value-preserving. Thus, Corollaries 4 and 5 follow from Theorem 2. □

# 4. INFORMATIVENESS

In this section we analyze the informativeness of the existing trust mechanisms as well as our new hybrids. A trust mechanism shall help agents to find good partners to interact with. Similar to ideas by Bolton et al. [5], we call a mechanism *informative* if it discriminates well between good and bad agents, and *non-informative* if it does not. A perfectly informative mechanism would be one that is perfectly discriminative in the sense that it has a strictly monotonic relationship between the trust scores $M_j(G, v_i)$ and the agent types $\theta_j$. With limited information, no mechanism can be perfectly informative and thus we want to measure how close our mechanism comes to this goal. We assume a linear relationship between agent types and trust scores. Then, the correlation between agents' types and the trust scores a mechanism produces tells us how discriminative the mechanism is. A random mechanism results in a correlation of 0. A perfectly discriminative mechanism results in a correlation of 1. Thus, all mechanisms that perform better than random have informativeness between 0 and 1. We define the informativeness of a mechanism $M$ on graph $G$ as the correlation between the true agent types and the trust scores produced by mechanism $M$. More formally, we offer the following natural definition:

DEFINITION 17 (INFORMATIVENESS). *Let $\Theta_{-i}$ denote the $(n-1)$-dimensional vector of all agents' types except for agent $i$. Let $\Theta_-^n = \langle \Theta_{-1}, \Theta_{-2}, ..., \Theta_{-n} \rangle$ denote the vector resulting from combining all $\Theta_{-i}$ vectors to a vector of dimension $(n-1)^n$. Given a trust graph $G = (V, E, w)$, and transitive trust mechanism $M$, let $M(G)$ denote the $(n-1)^n$-dimensional vector of all agents' trust scores from all other agents' perspectives produced by $M$, i.e., $M(G) = \langle M(G, v_1), M(G, v_2), M(G, v_3), ..., M(G, v_n) \rangle$. We define the informativeness of mechanism $M$ on graph $G$ as:*

$$\begin{aligned}
\mathit{Inf}(M, G) &= \mathit{correlation}(\Theta_-^n, M(G)) \\
&= \frac{\sum_{i=1}^n \sum_{j \neq i} (M_j(G, v_i) - \tilde{M})(\theta_j - \tilde{\theta})}{(n(n-1)-1) s_M s_\theta},
\end{aligned}$$

*where $\tilde{M}$ and $\tilde{\theta}$ are the sample means of the trust scores and the agent types; $s_M$ and $s_\theta$ are the sample standard deviations.*

## 4.1 Experimental Set-up

It is apparent from the definition that the informativeness of a mechanism is defined with respect to a particular trust graph $G$. Thus, to perform an informativeness measurement, we first have to specify how $G$ is generated in our experiments. In this section, we focus on a mechanism's ability to aggregate data and do not consider its strategyproofness. Thus, we will not consider strategic agents. Also, we want to measure informativeness independent from how the trust scores are being used by the agents when making decisions in the environment. Thus, we start our analysis with an artificial experiment where a random trust graph is constructed according to the following process.

We simulate a multi-agent system with 50 agents. Each agent's type $\theta_i$ is chosen uniformly at random from $[0, 1]$. In real-life networks, each agent will only have a small number of direct interactions relative to the total number of agents in the system. We model this in our simulation by limiting the maximum number of outgoing edges of all agents in the trust graph by $\kappa$. This "memory set" is selected uniformly at random for each agent at the beginning of the simulation. We let our simulation run for $\tau$ time steps. At each time step, each agent $i$ picks a random partner agent $j$ from its memory set. The outcome of the interaction between $i$ and $j$ is good with probability $\theta_j$ and bad with probability $1 - \theta_j$. Every agent keeps track of the total number of interactions and the number of successful interactions with each partner agent. At the end of each time step, for each agent $i$, we set the edge weight of edge $(i, j)$ equal to the fraction of successful interactions $i$ had with $j$ divided by the total number of interactions $i$ had with $j$. After $\tau$ time steps, we stop the interactive part of the experiment and consider the resulting trust graph $G$ as the basis for the analysis. For each mechanism $M$ that we consider, for each agent $i$ and each agent $j \neq i$, we compute the trust scores $M_j(G, v_i)$. We then calculate the informativeness metric, i.e., the correlation between the true agent types and the trust scores computed by the mechanisms.[3]

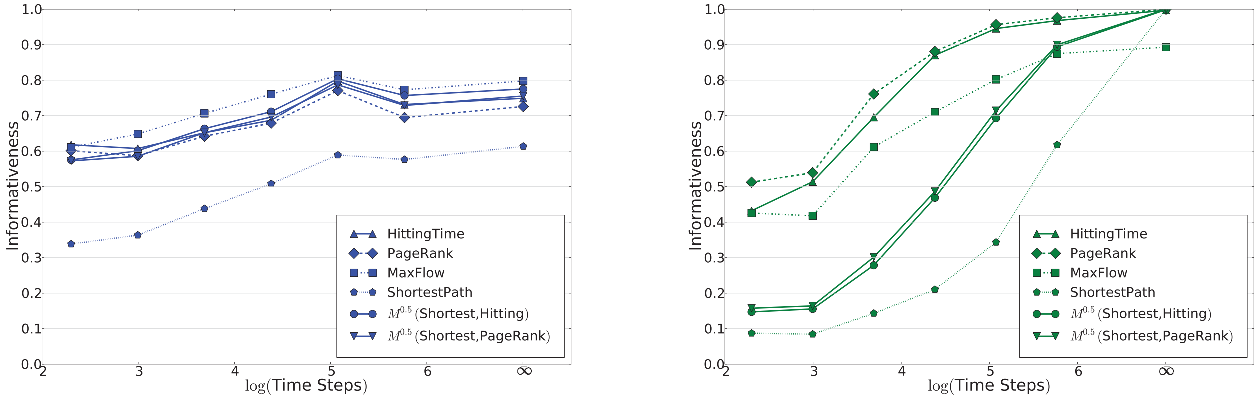
## 4.2 Informativeness of Existing Mechanisms

The informativeness metric is sensitive to the parameters of the trust graph generation process, in particular to the number of time steps, $\tau$, and to the size of the memory sets, $\kappa$. In Figure 2 we present two graphs that show some patterns that are representative for our experiments without strategic agents. For both graphs, we plot the log of the number of time steps on the x-axis, and the informativeness scores on the y-axis. Figure 2(a) shows results for $\kappa = 5$ and Figure 2(b) shows results for $\kappa = 50$, i.e., each agent interacts with every other agent in the system. The legend on Figure 2(a) holds for both graphs.

We see immediately that as the number of time steps increases, the informativeness scores increase for all mechanisms. This is expected because over time each agent gets better and better information about the type of each agent in its memory set. Note that the last data points in both graphs correspond to an infinite number of time steps. We simulated this by setting the edge weights for all agents inside the memory set equal to the true agents' types. It is interesting to note that all mechanisms, except for MaxFlow, reach informativeness of 1 when $\kappa = 50$ and $\tau = \infty$. However, for practical purposes this is less relevant, because in real-world trust graphs we will generally only have little information available.

In both graphs, we clearly see that the ShortestPath mechanism

[3] Note that this way, the informativeness score is already based on $50 \cdot 49 = 2,450$ trust score measurements. To remove noise, we run 5 trials, generating 5 graphs with the same parameters, increasing the number of trust scores to $12,250$ before computing the correlation.



(a) Maximum outdegree: $\kappa = 5$

(b) Maximum outdegree: $\kappa = 50$

**Figure 2: Informativeness experiment with 50 agents and uniform type distribution. We vary the number of time steps $\tau$.**

performs worst (except when $\tau = \infty$). This is expected and nicely illustrates the trade-off between informativeness and strategyproofness. The order of the other basic mechanisms is much less clear. In general, PageRank and HittingTime are close together, which makes sense given that both mechanisms use similar algorithms to compute trust scores. The MaxFlow mechanism shows the largest variation in informativeness and is particularly sensitive to $\kappa$, the size of the memory set. In Figure 2(a) where $\kappa = 5$, MaxFlow has the highest informativeness, while in Figure 2(b) where $\kappa = 50$, it has the second lowest informativeness. To explore this effect, we ran additional experiments for more values of $\kappa$ (not shown here). It turns out that an interesting cross-over effect happens at $\kappa = 10$: for $\kappa \leq 10$, the MaxFlow mechanism has informativeness as good as or better than PageRank and HittingTime, for $\kappa \geq 15$, MaxFlow has informativeness significantly worse than PageRank and HittingTime.

## 4.3 Informativeness of Hybrid Mechanisms

We now analyze the informativeness of two hybrid mechanisms: $M^\alpha$(Shortest, Hitting) and $M^\alpha$(Shortest,PageRank). We use these hybrids because the trade-off is clear in this case: ShortestPath has the best strategyproofness properties but the worst informativeness properties. We have shown analytically in the last section that the hybrids have intermediate strategyproofness properties and we expected the same result for informativeness.

Thus, it is perhaps surprising that, for many settings, the hybrids perform as well with respect to informativeness, or even better, than HittingTime or PageRank. In Figure 2(a), we see that $M^\alpha$(Shortest,PageRank) has informativeness scores that are as good or even higher than those of PageRank, and $M^\alpha$(Shortest, Hitting) has scores that are consistently higher than those of HittingTime. In contrast, in Figure 2(b), we see that both hybrids have intermediate informativeness, i.e., the informativeness scores of $M^\alpha$(Shortest, Hitting) lie between those of ShortestPath and HittingTime, and the scores of $M^\alpha$(Shortest,PageRank) lie between those of ShortestPath and PageRank. Further analysis (data not shown here) shows that another interesting cross-over effect happens: for large values of $\kappa$, both hybrids have intermediate informativeness as we expected. But for small values of $\kappa$, the informativeness of the hybrids is as good or even better than that of HittingTime or PageRank respectively. At first sight, it is counterintuitive that a hybrid mechanism could have informativeness even higher than any of its component mechanisms. A possible explanation is that both component mechanisms measure different aspects of the trust graph, and the hybrid mechanism benefits from both perspectives, i.e., both sources of information. A deeper analysis of this effect is subject of future research.

# 5. EFFICIENCY EXPERIMENTS

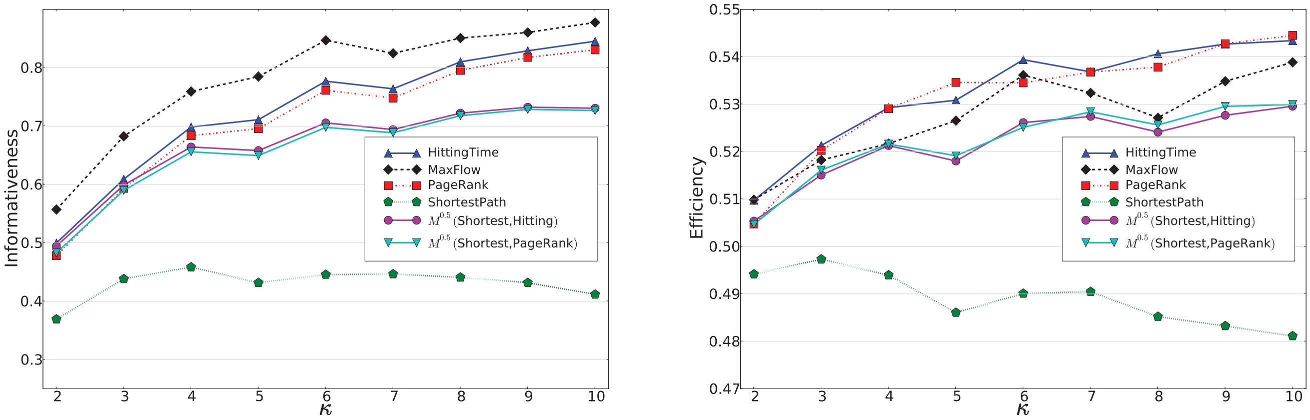
In this section we analyze the efficiency of hybrid mechanisms. We would like to investigate whether hybrids with intermediate informativeness and intermediate strategyproofness properties can achieve higher performance than any of the “pure” mechanisms. We measure the efficiency of a trust mechanism as the the fraction of transactions by non-strategic agents that are successful. Note that this is no longer independent of how agents use trust scores for acting in their environment. We consider two simulated domains: combatting the spread of bad files (e.g., viruses) in a file-sharing network, and ranking website quality based on link structure.

## 5.1 Experimental Set-up

Agents are divided into cooperative and malicious agents: cooperative agents have type $\theta_i = 0.95$, while malicious agents have types drawn uniformly at random from $[0, 0.5]$. A subset of the malicious agents are also strategic, i.e, they also consider manipulating the trust mechanism to their benefit.[4] We let $\gamma$ denote the fraction of the total agent population that is strategic. Properly simulating the behavior of strategic agents is difficult. We model strategic behavior by assuming a heterogenous fixed cost for manipulation (e.g., some agents are more adept than others at hacking the P2P file sharing software). As $\alpha$ increases, the manipulability of the mechanism increases linearly with $\alpha$, leading to higher rewards for manipulating agents. Since agents will only manipulate if the benefit exceeds their cost, we assume that the percentage of manipulating agents increases linearly with $\alpha$. For a manipulating agent, we determine in each context the optimal “attack” on the trust mechanism.

**Virus Distribution Experiment:** Imagine a file sharing network with good and malicious agents. Malicious agents have bad files that are infected by viruses. A trust mechanism helps to separate users with good files from users with bad files. In our experiments we use 100 simulated agents, of which 80% are malicious. We vary

[4] Note that the strategic agents are only *willing to consider* manipulating the trust mechanism. Whether they in fact perform manipulations depends on how costly the manipulations are and how much the agents can benefit from manipulating.



(a) Informativeness

(b) Efficiency

**Figure 3: Virus distribution experiment without strategic agents, varying the maximum outdegree $\kappa$.**

$\gamma$, i.e., the proportion of strategic agents, between 0 and 0.8. In P2P file sharing settings, the total number of agents in the system is too large for an agent to track all its interactions. Thus, we set the size of the memory set, $\kappa$, equal to 3. The memory set is selected uniformly at random for each agent at the beginning of the simulation.

We initialize the system by constructing a sparse trust graph. Each agent randomly chooses another agent $j$ from its memory set, and lays down an edge with weight 1 to $j$ with probability $\theta_j$. We repeat this process until each agent has exactly one outgoing edge. We then start the experiment itself and run it for 100 time steps. Each time step, agent $i$ obtains a set of three randomly selected agents drawn from the entire set of agents. With probability 0.9, the agent uses the trust mechanism to select agent $j$ with the highest trust score; with probability 0.1 the agent simply selects a random agent. This $\epsilon$-greedy selection policy encourages agents to explore and discover agents outside their memory set. Once $j$ is selected, with probability $\theta_j$, agent $j$ sends a good file (otherwise it sends a bad file). After the interaction has taken place, agent $i$ makes a report to the trust mechanism, updating the weight of its edge to agent $j$ to be the fraction of successful transactions over the total number of transactions. Strategic agents in this setting only employ misreport strategies because $M_\alpha(M_{\text{Shortest}}, M_{\text{Hitting}})$ is robust against sybil manipulations. By cutting all their out links they do not affect their own trust scores, but could lower the trust scores of agents ranked above them, thus improving their relative rank.

**Website Ranking Experiment:** This experiment uses a trust mechanism to rank websites according to their quality, helping web surfers differentiate between high quality and low quality websites. We assume that the set of surfers and the set of website owners coincides, i.e., each surfer has one pre-trusted website. We simulate 50 agents of which 80% are malicious (low quality websites) and we vary $\gamma$, the proportion of strategic agents (website owners), between 0 and 0.8.

We limit each agent to interacting with a randomly chosen memory set of size $\kappa = 5$. For each agent, we sample 10 times from that agent's memory set, simulate a transaction with each of the sampled agents, and finally update the edge weights as before corresponding to the number of successful interactions. Strategic agents (website owners) employ the misreport manipulation as well as a sybil manipulation (5 sybils) in the optimal star-shaped pattern [4].

We leave the trust graph unchanged over the duration of the experiment (i.e., surfers do not constantly update their own websites). We run the experiment for 100 time steps. At each time step, each surfer is provided with five randomly selected websites and considers their trust scores. We use a threshold-based selection rule: the surfer visits a random website from the set of websites with a trust score higher than a certain threshold (which we set to the median trust score across all agents).

## 5.2 Efficiency without Strategic Agents

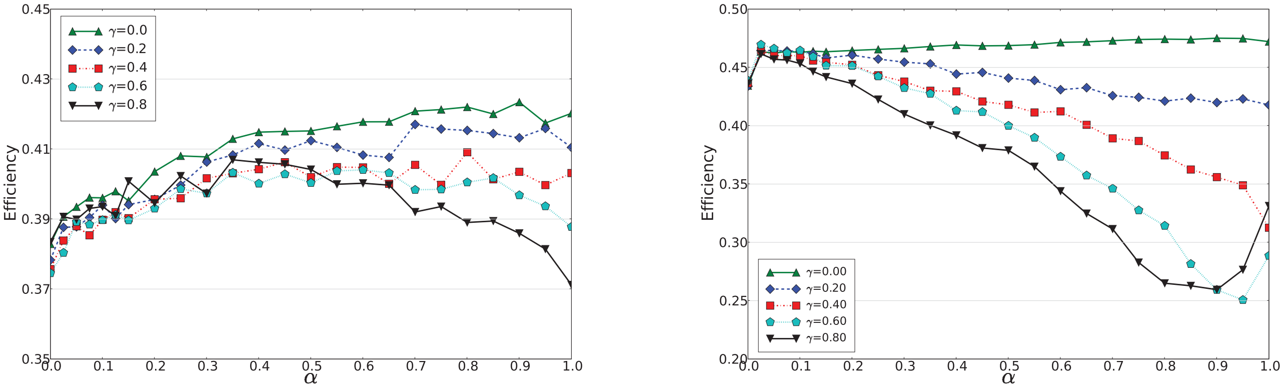
In Figure 3 we present efficiency and informativeness results (averaged over 10 trial runs) for the virus distribution experiment. In Figure 3(a), we plot the informativeness of the mechanisms on the y-axis, this time varying the maximum outdegree $\kappa$ on the x-axis. We see that the overall pattern is similar to the one we have described in Section 4. The ShortestPath mechanism has lowest informativeness and MaxFlow has highest informativeness. The mechanisms HittingTime and PageRank are close together and are slightly less informative than MaxFlow. We also see that the two hybrids $M^\alpha$(Shortest, Hitting) and $M^\alpha$(Shortest, PageRank) have intermediate informativeness.

Consider now Figure 3(b), where we plot overall efficiency on the y-axis and vary the maximum outdegree $\kappa$ on the x-axis. We see that the ordering of the mechanisms is the same as in Figure 3(a), except for the MaxFlow mechanism, which on average performs slightly worse than HittingTime and PageRank, even though it had better informativeness. Thus, without strategic agents and with the exception of MaxFlow, the informativeness of a mechanism seems to be a very good predictor of its efficiency. We have already seen in Section 4 that the MaxFlow mechanism is very sensitive to various parameter settings. A more detailed analysis of the properties of MaxFlow is subject of ongoing research.

## 5.3 Efficiency with Strategic Agents

We now analyze the efficiency of our hybrid mechanisms in the presence of strategic agents. In Figure 4(a) we display the results for the virus distribution experiment, and in Figure 4(b) the results for the WebRank experiment. On the x-axis we plot the blend factor $\alpha \in [0, 1]$ and on the y-axis we plot efficiency.

We see that with 0% strategic agents, efficiency increases almost monotonically as we move from ShortestPath to HittingTime or PageRank respectively. This is expected because ShortestPath is very uninformative, and without strategic agents, it has no benefits over the other mechanisms. However, the situation is different when strategic agents are present, i.e., for $\gamma \geq 0.2$. Now we see that for $\alpha$-values close to 1, the efficiency decreases significantly.



(a) Virus Distribution Experiment: $M^{\alpha}(Shortest, Hitting)$

(b) WebRank Experiment: $M^{\alpha}(Shortest, PageRank)$

**Figure 4: Efficiency analysis for hybrid mechanisms with strategic agents, varying blend factor $\alpha$.**

This is also expected because HittingTime and PageRank are both susceptible to the manipulations performed by strategic agents and thus, the more weight we give those mechanisms, the more successful the strategic agents are at manipulating the hybrids.[5]

The most important finding, however, is that initially, the efficiency goes up as we increase $\alpha$ and the efficiency peak in both cases does not occur for one of the base mechanisms. Instead, the efficiency peak in Figure 4(a) is around $\alpha = 0.5$ with a relative efficiency increase up to 5%. In Figure 4 the peak is around $\alpha = 0.02$ with a relative efficiency increase up to 7%. Thus, when strategic agents are present, the optimal hybrid mechanisms achieve higher overall efficiency than either of the component mechanisms.[6]

## 6. CONCLUSION

In this paper, we have introduced hybrid transitive trust mechanisms, which allow for a continuum of design tradeoffs between existing point solutions in the literature. We have shown analytically that these hybrids have intermediate strategyproofness properties. We have presented a simple metric to measure informativeness of trust mechanisms and via simulations we found that hybrid mechanisms have intermediate or sometimes even better informativeness than any of their component mechanisms. Finally, we have performed efficiency experiments to study the overall effect of using hybrid mechanisms. Our experimental results suggest that in some domains it is possible to improve efficiency by blending together two mechanisms, making a tradeoff between informativeness and strategyproofness that is optimal for a given population of agents. Note that the optimal $\alpha$ depends on the agent population and how costly it is for strategic agents to actually manipulate the mechanism. Our current experimental methodology is deliberately simplistic: as we increase blend factor $\alpha$ from 0 to 1, we also increase the fraction of strategic agents that choose to manipulate. This models a simple cost-benefit tradeoff. As a next step, we will instead assume a model in which this cost-benefit analysis is made explicit. In future work we will also consider the computational requirements of the trust mechanisms. For practical applications, informativeness and strategyproofness are important, but in any case it must be feasible to run the mechanisms on real-sized graphs.

## 7. REFERENCES

[1] A. Altman and M. Tennenholtz. On the Axiomatic Foundations of Ranking Systems. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, August 2005.

[2] A. Altman and M. Tennenholtz. Quantifying Incentive Compatibility of Ranking Systems. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, Boston, MA, July 2006.

[3] A. Altman and M. Tennenholtz. An Axiomatic Approach to Personalized Ranking Systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, January 2007.

[4] M. Bianchini, M. Gori, and F. Scarselli. Inside Pagerank. *ACM Transactions on Internet Technology*, 5(1), 2005.

[5] G. Bolton, B. Greiner, and A. Ockenfels. Engineering trust - reciprocity in the production of reputation information. Working Paper No 42, University of Cologne, Dept. of Economics, 2009.

[6] A. Cheng and E. Friedman. Sybilproof Reputation Mechanisms. In *Proceedings of the ACM Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, Philadelphia, PA, August 2005.

[7] A. Cheng and E. Friedman. Manipulability of PageRank under Sybil Strategies. In *Proceedings of the First Workshop on the Economics of Networked Systems (NetEcon)*, Ann Arbor, MI, June 2006.

[8] E. Friedman, P. Resnick, and R. Sami. Manipulation-resistant reputation systems. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 27. Cambridge University Press, 2007.

[9] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International World Wide Web Conference*, New York, NY, May 2004.

[10] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[11] P. Resnick and R. Sami. Sybilproof transitive trust protocols. In *Proceedings of the 10th ACM Conference on Electronic Commerece (EC)*, Stanford, CA, July 2009.

[12] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on eBay: A controlled experiment. *Experimental Economics*, 9:79–101, 2006.

[13] D. Sheldon and J. Hopcroft. Manipulation-Resistant Reputations Using Hitting Time. In *Proceedings of the 5th Workshop on Algorithms for the Web-Graph*, San Diego, CA, December 2007.

[5] Note that in Figure 4(b), for $\gamma = 0.6$ and $\gamma = 0.8$, the efficiency increases again as we move from $\alpha = 0.9$ to $\alpha = 1$. This happens because at $\alpha = 0.9$, the strategic agents affect the hybrid twice, via ShortestPath and via PageRank. As we have seen in Figure 2, ShortestPath is particularly bad when it has little information. For $\alpha = 0.9$, the strategic agents cannot influence their trust scores under ShortestPath, but the mechanism still stuffers significantly from the missing information due to many misreport attacks. Close to $\alpha = 1$, ShortestPath loses effect, and as we have seen in Figure 2, PageRank is significantly better at coping with little information in the trust graph which explains the efficiency increase at the end.

[6] We also measured the informativeness for the two experiments with strategic agents, varying the blend factor $\alpha$. For the virus distribution experiment, the best hybrid has informativeness that is 14% higher than that of ShortestPath or HittingTime. For the WebRank experiment, the best hybrid has informativeness that is 10% higher than that of ShortestPath or PageRank.