Multi-Armed Bandit Policies for Reputation Systems

Thibaut Vallée, Grégory Bonnet, and François Bourdon

Normandie Université, France
UNICAEN, GREYC, F-14032 Caen, France
CNRS, UMR 6072, F-14032 Caen, France
firstname.lastname@unicaen.fr

Abstract. The robustness of reputation systems against manipulations have been widely studied. However, the study of how to use the reputation values computed by those systems are rare. In this paper, we draw the analogy between reputation systems and multi-armed bandit problems. We investigate how to use the multi-armed bandit selection policies in order to increase the robustness of reputation systems against malicious agents. To this end, we propose a model of an abstract service sharing system which uses such a bandit-based reputation system. Finally, in an empirical study, we show that some multi-armed bandits policies are more robust against manipulations but cost-free for the malicious agents whereas some other policies are manipulable but costly.

1 Introduction

In a multi-agent system, when an agent cannot carry out a task alone, it needs to delegate it to another agent. In such systems, agents need to share skills and knowledge, and thus agents are both service consumers and providers. However, as large open multi-agent systems allow heterogeneous agents to interact, some agents can provide bad quality services due to computation or network failures, or even due to malicious behaviours. For instance, such problems as corrupted files (failures) and viruses (malicious behaviours) spreading are common in peer-to-peer file sharing systems (as Gnutella [1]). A common way to help agents to select with whom they will interact[1] is to use a reputation system. Such systems allow agents to ask services to other agents whom have been advised by a third-party. Agents evaluate their past interactions and compute a value which represents how they trust each other agent with whom they have interacted. These trust values are communicated by the agents and agregated through feedbacks. Then these feedbacks are used to compute a reputation value for each agent, that is assumed to reflect their reliability as service providers. Many reputation systems have been proposed but, in those systems, a malicious agent can lie, collude with other agents, introduce many false identies called Sybil agents, leave and join the system with a new identity, or change its behaviour in order

[1] We say that two agents interact when one provides a service to the other.

to manipulate its reputation value. Several works propose reputation systems which are robust to a specific manipulation. However those studies focus on how the trust and reputation values are computed but not on how the agents will use it. Indeed, the policy used to select providers impacts the system. If each agent interacts only with the one which has the best reputation value, it will be hard for a single malicious agent to provide many bad services. However, such a policy leads few service providers to be overloaded while other providers never interact. Conversely, if an agent selects randomly with whom it will interact, the system opens but the reputation value is useless: malicious agents can easily provide bad services. Moreover, many reputation systems are robust to one-shot manipulations but sensitive against collusions of agents that execute a long-term manipulation. For instance, on eBay [2], an agent can behave in a good way for many low-priced transactions in order to increase its reputation value and can behave badly for rare high-priced transactions. The problem of selecting with whom interacting based on past observations has been widely studied in the context of multi-armed bandit (MAB). In this paper, we propose to investigate how using the MAB policies in a reputation system can decrease the number of manipulations efficiently. Our work is organized as follows. We present in Section 2 the literature in the field of reputation systems, their manipulations and the field of MAB. In Section 3, we propose a model of service sharing system and draw the analogy between this model and the Multi-Armed Bandit problem. We present in Section 4 some canonical policies and manipulations. Finally, we present in Section 5 an empiric study of the performance of the system when a coalition of agents tries to manipulate it.

2 Related work

Trust was introduced by Marsh [3] in the context of multi-agent systems. This notion formalizes an estimation of the future behaviour of an agent when there exists a risk of unexpected behaviour. Three fundamental axioms define what a reputation system is [4]: (1) the agents in the system will interact in the future; (2) feedbacks, called trust values, on the interactions between agents must be shared with the other agents; (3) those feedbacks must be used to help consummers to decide which will be their next providers. Thus, in reputation systems, the trust value of an agent about another is the evaluation of the past interactions by the former about the latter. Then, the reputation of an agent is an agregation of all the trust values about this agent. Many reputation systems have been proprosed [4–11]. They can be classified in three families: symmetric (e.g. eBay's reputation system [4]), assymmetric global (e.g. Google's Page Rank) and assymmetric personnalized (e.g. maxflow-based algorithm). Two of the more common reputation systems are BetaReputation [6] and EigenTrust [7]. BetaReputation uses a Beta density function to compute the probability that an agent exhibits a good behaviour. EigenTrust uses the same algorithm than Google's Page rank: given a graph which represents the trust values between the agents, the reputation of an agent is the probability than a random walker passes by the node corresponding to this agent. Let us notice that EigenTrust is known to be manipulable by a simple coalition of agents [12]. The problem of the robustness of reputation systems has been strongly studied [10, 11, 13]. Cheng and Friedman [13] proved that no symmetric reputation system can be robust to false-identity collusions and only assymetric reputation systems can be robust if they satisfy some strong conditions. Altman *et al.* [11] defined, among other axioms for reputation and ranking systems, the incentive compatibility which corresponds to a robustness against manipulation, and they proved that most of the ranking systems do not satisfy it. However, both Cheng and Friedman, and Altman *et al.* considered manipulations at a given instant: a system is robust if the manipulation does not change the reputation value (or rank) at the time the manipulation is perfomed. They do not investigate if it is possible to manipulate the reputation system over time. Indeed, some manipulations as strategic oscillation [14] are built to manipulate the reputation systems on a long term. Moreover, most of those papers consider specific manipulations but do not study how using the reputation values to select the most reliable agents can impact the system robustness. Pinyol and Sabater [15] highlighted the notion of learning/adaption strategy which is how the agents use the reputation to adapt their behavior for future interactions. Although most of the reputation systems do not offer clear strategies, a similar problem of selection has been studied in another context: the multi-armed bandit problem (MAB) [16]. The canonical definition of this problem is the following. Let us consider a gambling machine with multiple arms. Each arm has an unknown reward function. Thus, the problem is which arm an agent needs to pull in order to maximize its reward? Many models of MAB have been studied (for instance with multiple players [17], stochastic or stationary policies [18]). All these models propose selection policies to minimize the agent's regret: the difference between the reward it obtained and how much it could had won if it had always pulled the best arm. All this policies, such as UCB, Poker, ε-greedy [19, 20], are a compromise between pulling the arm which has the best expected reward and pulling another arm in order to increase the agent's knowledge on the reward distributions (known as the exploration - exploitation compromise). In this paper, we propose to draw an analogy between both problems: the selection of agents evaluated by a reputation value and the selection of arms evaluated by an estimated reward function. We investigate how using MAB policies in a reputation system impacts of the manipulations, which had not been studied to the best of our knowledge.

3 A general model using reputation system

The aim of a reputation system is to help each agent to determinate with which agent it will interact in order to achieve its goal. In this section, we propose a general application where the agents must interact with the others and use a reputation system. In such system, the agents use a policy in order to select with whom interact. By analogy with the multi-armed bandit problem, we propose to used the MAB policies in such system.

3.1 A service sharing system model

Considering a multi-agent system where each of them can provide some services. In order to be general, we consider abstract services. A such system is called a *service sharing system*: when an agent needs a service that it cannot provide itself, it ask this service to another agent.

Definition 1. *A service sharing system is a tuple $\langle N, S \rangle$ where N is a set of agents and S a set of available services. We denote by $N_x \subseteq N$ the set of agents that can provide the service $s_x \in S$.*

Definition 2. *In a service sharing system, an agent $a_i = \langle \overrightarrow{\varepsilon_i}, v_i, T, f_i, \pi_i \rangle$ is an entity which can consume and provide services where: $\overrightarrow{\varepsilon_i}$ is its expertise vector; v_i is its evaluation function; T is the matrix of trust values; f_i is its reputation function; π_i is its policy.*

The *expertise* of $a_i \in N$ for the service $s_x \in S$, denoted $\varepsilon_{i,x}$, is the capacity for a_i to performs s_x with a good quality when another agent asks it to. Even if the quality of a service depends on the expertise of the provider, it is subject to the consumer evaluation. This evaluation can be based on many factors. For instance, in peer-to-peer file sharing systems, the quality can be evaluated on the download latency, the file quality and so on, such as it can take many kind of values: booleans, $[-1; 1]$, $\mathbb{N}$, $\mathbb{R}$ or any other representation. In order to stay general, we assume that for all agents $a_i \in N$, a_i evaluates the services with its *evaluation function* $v_i : S \rightarrow V$ where V is a common codomain for all agents. We assume that the agents agregate their past experiences in a trust matrix (denoted T) and use feedbacks to share with the others their observations. The agents can provide a feedback each time they receive a service, or only when it is necessary to avoid communication flooding. The trust value of the agents represents only how each agent evaluates the service that it received from the others. The reputation of an agent a_i is the agregation of all the local trust values about a_i. In this article, we do not focus ourselves on how the reputation is computed. We only assume that each agent uses a *reputation fonction* $f_i : N \times S \rightarrow \mathbb{R}$. Hence each agent can compute alone with its knowledge of T the reputation of the other agents for each service. We make no assumption on the reputation function and allow two agents to use different reputation functions. The reputation of the agents is assumed to represent if they can provide a given service with a good quality. The *policy* of the agent $a_i \in N$ defines how it uses those reputations in order to select an expected good service provider: $\pi_i : S \rightarrow N_x$. We do not make any assumption on how the policy is computed and allow the agents to follow differents policies. The Figure 1 resumes the different interactions between agents in this application. The arrow 1 represents service requests from a_i to other agents (selected by the policy). The arrow 2 represents this service as provided. On the other side, arrows 3 and 4 are respectivly service requests from an agent a_j to a_i and the service that a_i provides to a_j. We represent the feedbacks by arrows $5, 6, 7$ and 8 (respectivly a feedback request from a_i to a_j, a feedback answer from a_j to a_i, a feedback request from a_j to a_i and a feedback answer from a_i to a_j).

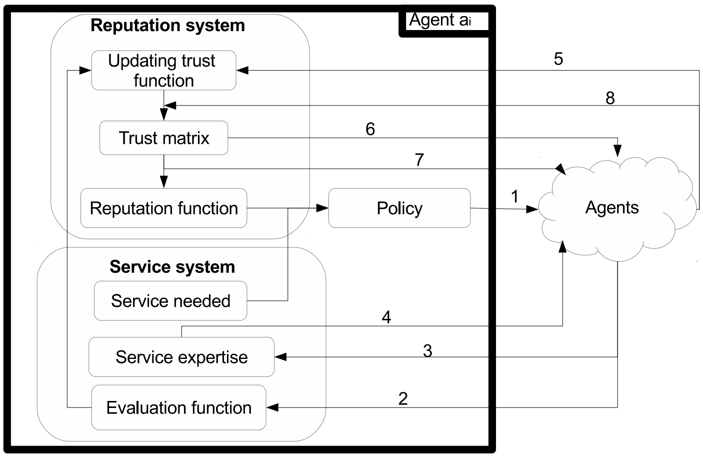

Fig. 1. Interactions between agents in the service sharing system

3.2 Analogy with multi-armed bandit problems

The aim of the policy in the service sharing system is to determinate to which agent asking a service. Such problem is related to a multi-armed bandit problem. Let us consider a player and a gambling machines with multiple arms. Each of these arms has an unkown reward function. The problem is which arm the player needs to pull in order to obtain the best possible reward? Both problems, service sharing and multi-armed bandit, use past observations to estimate the future service quality/reward of an agent/arm if it is selected/pulled. Thus, we can modelise a service sharing system with a MAB where each agent is in the same time a player and a gambling machines, and each arm corresponds to a service that the agent can provide.

Definition 3. *Let a $\langle N, S\rangle$ be a service sharing system. The corresponding MAB is defined by the set of M multi-armed bandits where $|M| = |N|$ and $\forall a_i \in N, \forall s_x \in S : a_i \in N_x$, there exists one and only one arm $m_{i,x}$ on the slot machine m_i. The expected reward of the arm $m_{i,x}$ is $\varepsilon_{i,x}$.*

In this MAB, agents communicate to share their observations. A such exchange of knowledge allows the agents to use the past experiences of the others in order to approximate the expected the reward of each arm. However, some feedbacks can be deceitful. The reputation system in this MAB helps the agents to agregate theirs observations. In this context, the agents can compute a reputation value for each arm. This value does not correspond exactly to the expected reward. Indeed, if an agent uses EigenTrust as reputation function, the reputation of an arm is the ratio of reward that it had provided on

Table 1. Analogy between the service sharing system and MAB

	Service sharing system	**MAB**
Aim	Maximize the services quality	Maximize the reward
Actors	Agents (consumers) Agents (providers)	Players Bandits
Interactions	Asking a service	Pulling an arm
Capacity	Expertise	Reward distribution function
Gain	Service quality	Reward
Observations	Trust matrix	Past observations
Communication	Feedback on another agent	Feedback on a arm
Reputation	Expected behaviour	Expected reward
Policy	Gives the next service provider	Gives the next arm to pull
Manipulations	Malicious agents	Adversarial players

the sum of all reward. However, we assume that for two arms $m_{k,x}$ and $m_{k',x}$, $f_i(m_{k,x}, s_x) > f_i(m_{k',x}, s_x)$ implies that the expected reward of the first is better than the reward of the second. As both are correlated (the arm with the best reputation is the one with the best expected reward), we consider that reputation of an arm is an approximation of the expected reward. Table 3.2 sums up the analogy between the services sharing system and a MAB. Based on this analogy, we propose to use canonical policies of multi-armed bandit problems in a service sharing system.

4 Agents strategies

In this section, we define firstly the MAB policies in our model. Secondly, as some malicious agents can try to manipulate this system, we define some threats models. Finally, in order to evaluate the impact of the policies against this manipulations, we define several performance metrics.

4.1 Policies from multi-armed bandit problem

To resolve the multi-armed bandit problem, many solutions have been studied [21, 22]. We adapt two of them, UCB and ε-greedy policies, and propose a third: the ε-elitist policy. All of them make a compromise between optimizing the reward and exploring the system in order to refine the agent's knowledge.

The main algorithm to solve MAB problems is UCB (Upper Confidence Bound). UCB allows the agent to select another machine than the one which has the best expected reward in order to increase its knowledge about the system. We recall we assume that the reputation of an agent is an approximation of the expected quality of a service that it can provide.

Definition 4. *An agent follows* UCB policy *if it selects the agent $a_j \in N_x$ which maximizes $f_i(a_j, s_x) + \sqrt{\frac{2\ln(1+n_x)}{1+n_{j,x}}}$ where $n_{j,x}$ is the number of services s_x that has provided a_j to a_i and n_x is the number of services s_x that a_i has received.*

An intuitive policy for an agent is to ask services to the agent which has the best reputation value. Such policy is called *elitism* and the agent which has the best reputation value will be always solicited. Another trivial policy called *uniform policy* consists in selecting a_j uniformly at random in N_x, and to not use the reputation of the agents. Thus, we propose to use the ε-greedy policy [21] that is a mixed policy between elitism and uniform policy.

Definition 5. *An agent $a_i \in N$ follows an* ε-greedy policy *if it selects the provider $a_j \in N_x$ which have the best reputation value with a probability of* $1 - \varepsilon$ *and, with a probability ε, it selects a provider uniformly at random in N_x.*

Notice that if $\varepsilon = 0$ this policy is elitism, and if $\varepsilon = 1$ the policy is uniform. We propose also a third policy called the ε-elitism policy. Intuitively, an agent which follows this policy selects the future provider randomly within the $\varepsilon \times |N_x|$ agents which have the best reputation values.

Definition 6. *Let $N'_x \subseteq N_x$ such that $|N'_x| = \lceil \varepsilon \times |N_x| \rceil$ and that $\forall a_j \in N'_x, \nexists a_k \in N_x \setminus N'_x : f_i(a_j, s_x) < f_i(a_k, s_x)$. An agent $a_i \in N$ follows an ε-elitist policy if it selects uniformly at random a_j in N'_x.*

4.2 Threat model

As we intend to investigate the policies robustness to malicious behaviors, we assume firstly that an agent is honest if the quality of its services are in accordance with its expertise vector and if its feedbacks about another agent are its trust value about this latter. In opposite, we define a malicious agent as an agent which provides willingly a service with a bad quality or gives a false feedback about an agent. We make two assumptions on the malicious agents in our system. Firstly, all malicious agents are in a coalition (denoted $\mathcal{M} \subset N$) as if it exists two coalitions, both coalitions try to manipulate the other as if it is composed of honest agents. Secondly, they aim at maximizing the number of bad services that they provide as if a malicious agent a_i provides only good services, the agents which interact with are satisfied and a_i cannot be considered as malicious. Remark we consider coalitions as reputation systems are robust to single malicious behaviours but still vulnerable to collusion [10]. Moreover, any single malicious agents can use false identities (called Sybil [23]) in order to form a coalition with itself. It exists many manipulations as slandering, promotion, withewashing [10] that aim at modifying the malicious agents' reputation values. Those manipulations can be applied in a single timestep. Moreover, some manipulations as the oscillating manipulation apply over time. In order to consider the worst possible setting, we agregated slandering, promotion, withewashing and oscillating manipulation in a single malicious behaviour. Let a malicious coalition $\mathcal{M}$ which is splitted in two subsets $\mathcal{M}_1$ and $\mathcal{M}_2$. At each timestep, the malicious agents apply the following strategy:

- the agents of $\mathcal{M}_1$ slander the agents of $N \setminus \mathcal{M}$;
- the agents of $\mathcal{M}_2$ promote the agents of $\mathcal{M}_1$;
- the agents of $\mathcal{M}_1$ provide willingly "bad" services;
- the agents of $\mathcal{M}_2$ provide their services with respect of their expertise factor;
- when $a_i \in \mathcal{M}_1$ has a low reputation value, it whitewashes. An agent of $\mathcal{M}_2$ changes its behaviours and joins $\mathcal{M}_1$ and the new identity a_{n+1} joins $\mathcal{M}_2$.

When a coalition of agents manipulate the system, they impact the performance of the system. Thus, we define how to evaluate this impact.

4.3 System evalutation

In order to evaluate the performance of those policies, we propose some metrics of performance. A common metrics for MAB is the *regret* [21, 22]. Intuitivly, the regret of an agent is the difference between the reward that it could have won if it had interacted with the provider whom had the best reputation and the reward that it has obtained. The aim of our model is to maximize the number of good services provided. Thus, we define the system efficiency as the complementary of the regret.

Definition 7. *Let R_i be the set of services that have received the agent a_i and let R_i^+ be the set of good services that it received. The efficiency of the system is the ratio:* $\sum_{a_i \in N} |R_i^+| \; / \sum_{a_i \in N} |R_i|$

In opposite, the malicious agents search to maximize the number of bad services that they provide. However, manipulating the system has a cost for the malicious agents. Indeed, in order to maintain a good reputation, the agents provide sometimes good services that is in opposite to their goal. We define hence a malicious cost measure.

Definition 8. *Let P_i be the set of services that the agent a_i has provided and let P_i^+ be the set of good services that it has provided. The manipulation cost is the ratio:* $\sum_{a_i \in M} |P_i^+| \; / \sum_{a_i \in N} |P_i|$

As we consider open multi-agent systems, some policies, such as the elitism, make that a small subset of the agents will provide the services, and thus those agents can be overloaded. Moreover, only this subset of agents will see their reputation value updated. As in [7] we mesure the load balancing in the system.

Definition 9. *Let $N_t \subseteq N$ be the subset of agents that have provided services at the timestep t. The load balancing is the ratio:* $|N_t| \; / \; |N|$

Those three metrics are defined in order to evaluate the robustness of the system against a malicious coalition. The system efficiency defines how much the malicious agent provide bad services. The manipulation cost represents how much the malicious agents must pay in order to manipulate. The load distribution represents how the policy impacts the openness property of the system.

5 Experiences

In this section, we evaluate the policies against a malicious coalition. To the best of our knowledge, there is no other works to compare with as we do not evaluate the reputation systems but the policies that use such systems.

5.1 Protocol

For simplicity, we assume that only one service is provided: sharing a file. At each timestep, each agent asks to another agent a file that it does not have. We also assume that providing a file is completed in a single timestep. We do not limit the number of files that can be provided by an agent in one step. The expertise of the agents is drawing uniformly at random. As in our model, we make no assumption on the reputation system used, we study here our policy on two canonical reputation system: EigenTrust [7] and BetaReputation [6]. We assume that the agents detect immediatly if the file they received is good or not. We investigate the uniform, UCB, ε-greedy and ε-elitist (with $\varepsilon \in [0;1]$) policies. In these experiences, we consider a coalition of malicious agents which applies the thread model given in Section 4.2. We initialize the simulations with 100 agents which interact during 100 timesteps. At each timestep, we consider that it has a probability of 0.01 that an honest agent joins or leaves the system in order to simulate an open system. At $t = 100$, we introduce 10 malicious agents which try to manipulate the system during 1000 timesteps in order to simulate a malicious coalition trying to manipulate a running system. We reiterate those simulations 50 times and compute the average metrics with their 95% confidence intervals. Although, all the results are dependent on a huge number of parameters, we claim these results give us insights about the policies distinctive features. For instance, increasing the number of malicious agents simply decreases the system efficiency and increases the manipulation cost (all other things being equal).

5.2 Results and analysis

For readability, we present only four policies: uniform, UCB, 0.2-elitist and 0.2-greedy. The uniform policy is used as a baseline. The main result of this empirical study is that the policy used influences the robustness of the reputation system against manipulations. UCB is clearly sensitive to a strategic manipulation but is costly for the malicious agents. In the over side, the robustness of a reputation system which uses a ε-greedy policy depends essentially on the robustness of its reputation function. Morever, even if the malicious agents provides a small set of bad services, manipulating the ε-greedy policy is costless. Finally, using a ε-elitist policy is a compromise between UCB and ε-greedy policy. Figure 2 shows the system efficiency under the policies. As we can see, the UCB policy is clearly sensitive to a malicious coalition, even with a BetaReputation wich is more robust to the manipulation than EigenTrust. In the other side, the 0.2-greedy policy is robust to the manipulations on BetaReputation system but not on EigenTrust. As it is a manipulable reputation function, the malicious agents can easily have

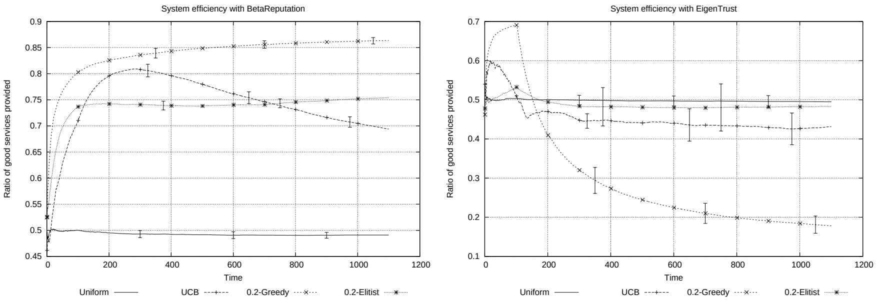

Fig. 2. System efficiency

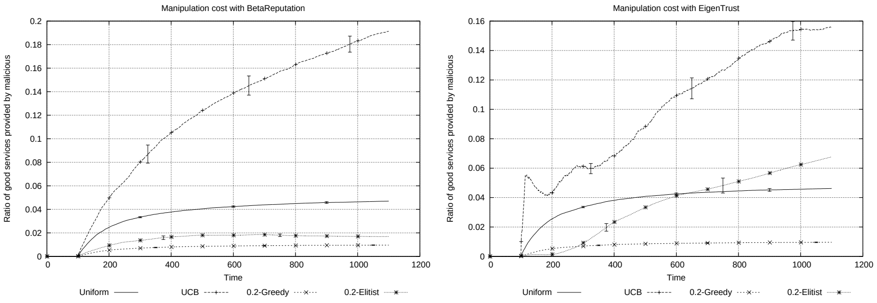

Fig. 3. Manipulation cost

a good reputation value and the greedy policy selects them. Hence, we assume that the robustness of the greedy policy is linked to the reputation function used, which is not the case for UCB. Denote that the 0.2-elitist policy is less effective than the 0.2-greedy with BetaReputation system but less manipulable with EigenTrust. UCB policy is clearly manipulable. However, the Figure 3 shows us that UCB is also costly for the malicious agents. In order to maintain a good reputation values, the malicious agents must provide more good services than bad services. Indeed, the manipulability of UCB comes from the fact that it selects the providers on whom the consumer has the least knowledge. Hence, in order to manipulate the system, the malicious agents need to frequently whitewash which is very costly. On the other side, the 0.2-greedy policy is almost cost-free for the malicious agents. In EigenTrust, the malicious agent can provide a large number of bad services without providing good services in order to increase their reputations values. The 0.2-elitist policy is a compromise between manipulation efficiency and cost: the malicious agents can provide bad services but they must provide good services too. The load balancing presented in Figure 4 shows us the degree of openness of the system. Remark that the greedy policy always selects a small subset of agents. As this policy selects the agents with the best reputation values, the probability for a new agent to be selected is small. Hence,

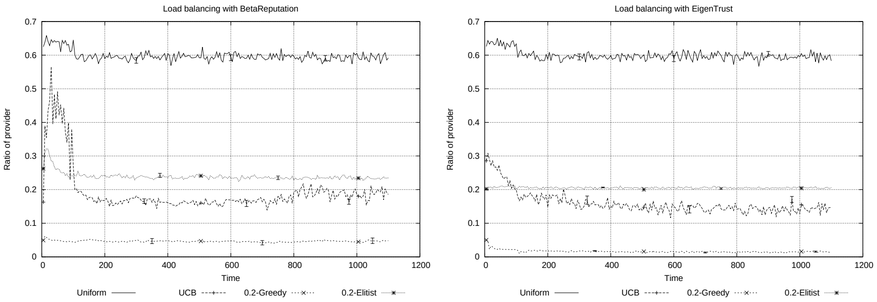

Fig. 4. Load balancing

using a greedy policy implies that new agent cannot be selected. Thus, this policy is effective against whitewashing but at the cost of the openness of system. Moreover, if a malicious agent manages to have a better reputation value than honests agents (for instance promotion and slandering with EigenTrust), this malicious agent is always selected and hence can provide bad services. UCB selects a greater subset of providers. Indeed, this policy allows the agent to explore the agents that they do not know. Hence UCB is more sensitive to whitewashing but also more open than greedy policy. To conclude this empirical study, UCB is manipulable but also very costly for the malicious coalition, and the robustness of a services sharing system which uses a ε-greedy policy depends on the robustness of its reputation function. Moreover, manipulating such policy is almost cost-free. A ε-elitist policy is a compromise between robustness and cost of the manipulation. We show also that the robustness against whitewashing has a cost on the openness property of the system.

6 Conclusion

In this paper, we propose a model for service sharing system which combines reputation systems and selection policies. As the problem of selection policy in services sharing systems and in multi-armed bandits are closely related, we propose to use multi-armed bandits policies in the service sharing system in order to fight against malicious agents. We study empiricaly the impacts of canonicals policies on manipulations. These policies are either sensitive against manipulations but costly for the malicious agents, or dependent on the reputation function robustness but almost cost-free. Finding a selection policy which is in the same time robust against manipulations, costly for the malicious agents and that does not impact the openness of the system is still an open problem. In a future work we intend to modelise a reputation multi-armed bandit where feedbacks could be seen as pulling a specific arm of a bandit. Moreover, we expect to clearly distinguish the trust in the expertise and the trust in the feedbacks. As there is no reputation function robust against all manipulations, we propose to aggregate several reputation functions in order to increase the robustness. A such problem has been considered on the multi-armed bandit problem by Auer [18] where players have a set of policies for choosing the best action.

References

1. Adar, E., Huberman, B.A.: Free riding on Gnutella. First Monday (2000)
2. Dini, F., Spagnolo, G.: Buying reputation on eBay: Do recent changes help? IJEB (2009) 581–598
3. Marsh, S.P.: Formalising trust as a computational concept. PhD thesis, University of Stirling (1994)
4. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. ACM Communications (2000) 45–48
5. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the Web. Technical report, Stanford InfoLab (1999)
6. Jøsang, A., Ismail, R.: The Beta reputation system. In: 15th BledEC. (2002) 41–55
7. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The EigenTrust algorithm for reputation management inP2P networks. In: 12th WWW. (2003) 640–651
8. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision support systems (2007) 618–644
9. Rahbar, A., Yang, O.: PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing. IEEE PDS (2007) 460–473
10. Hoffman, K., Zage, D., Nita-Rotaru, C.: A survey of attack and defense techniques for reputation systems. CSUR (2009)
11. Altman, A., Tennenholtz, M.: An axiomatic approach to personalized ranking systems. JACM (2010)
12. Cheng, A., Friedman, E.: Manipulability of PageRank under Sybil strategies. In: 1st NETECON. (2006)
13. Cheng, A., Friedman, E.: Sybilproof reputation mechanisms. In: 3rd P2PECON. (2005) 128–132
14. Srivatsa, M., Xiong, L., Liu, L.: TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks. In: 14th WWW. (2005) 422–431
15. Pinyol, I., Sabater-Mir, J.: Computational trust and reputation models for open multi-agent systems: a review. Artificial Intelligence Review (2013) 1–25
16. Robbins, H.: Some aspects of the sequential design of experiments. Bulletin of the AMS (1952) 527–535
17. Liu, K., Zhao, Q.: Distributed learning in multi-armed bandit with multiple players. IEEE SP (2010) 5667–5681
18. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: Gambling in a rigged casino: the adversarial multi-armed bandit problem. In: 36th FOCS. (1995)
19. Vermorel, J., Mohri, M.: Multi-armed bandit algorithms and empirical evaluation. In: 16th ECM. (2005) 437–448
20. Auer, P., Ortner, R.: UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. Periodica Mathematica Hungarica (2010) 55–65
21. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Machine Learning (2002) 235–256
22. Wang, Y., Audibert, J.Y., Munos, R.: Algorithms for infinitely many-armed bandits. NIPS (2008) 1729–1736
23. Douceur, J.R.: The Sybil attack. In: 1st IPTPS. (2002) 251–260