Trust-based Incentive Mechanisms for Community-based Multiagent Systems

by

Georgia Kastidou

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this thesis we study peer-based communities which are online communities whose services are provided by their participant agents. In order to improve the services an agent enjoys in these communities, we need to improve the services other agents offer. Towards this goal, we propose a novel solution which allows communities to share the experience of their members with other communities. The experience of a community with an agent is captured in the evaluation rating of the agent within the community, which can either represent the trustworthiness or the reputation of the agent. We argue that exchanging this information is the right way to improve the services the agent offers since it: i) exploits the information that each community accumulates to allow other communities to decide whether to accept the agent while it also puts pressure on the agent to behave well, since it is aware that any misbehaviour will be spread to the communities it might wish to join in the future, ii) can prevent the agent from overstretching itself among many communities, since this may lead the agent to provide very limited services to each of these communities due to its limited resources, and thus its trustworthiness and reputation might be compromised.

We study mechanisms that can be used to facilitate the exchange of trust or reputation information between communities. We make two key contributions. First, we propose a graph-based model which allows a particular community to determine which other communities to ask information from. We leverage consistency of past information and provide an equilibrium analysis showing that communities are best-off when they truthfully report the requested information, and describe how payments should be made to support the equilibrium. Our second contribution is a promise-based trust model where agents are judged based on the contributions they promise and deliver to the community. We outline a set of desirable properties such a model must exhibit, provide an instantiation, and an empirical evaluation.

Acknowledgements

If someone were to ask me to describe my PhD experience, I would simply say: ”wow, what an adventure!” A number of people have contributed to this adventure who I wish to thank, but first, I would like to thank Canada for giving me the opportunity to pursue my PhD studies here, meet new people, make new friends, be exposed to a multicultural environment, and for preparing me to teach younger people through these experiences. I would especially like to thank Dr. Jay Black who selected me from a large number of applicants and gave me the opportunity to come here.

During my PhD studies, I was fortunate to work with two supervisors: Dr. Kate Larson and Dr. Robin Cohen. I would like to express my deep and sincere gratitude to Dr. Kate Larson. Kate was initially on my PhD Advisory Committee. Very early my research swift under Kate's research areas. Even though I was not her student at that time, Kate stood by me, supported me, and believed in my work as if I was already one of her students. Kate has been a great advisor and mentor, and was always there for me, sharing the good moments, as well as the disappointments and frustrations I went through as a PhD student. Although Kate is in the beginning of her career, the undoubtedly proven and profound quality of her research, her exceptional stewardship, and her values make her the best Professor and advisor I have met. I am really proud to have had Kate as my supervisor.

I would also like to thank Dr. Robin Cohen for her guidance, support, insight, and financial assistance throughout the course of my graduate program. Dr. Cohen provided me with valuable advice throughout my academic career. My collaboration with Dr. Cohen made an invaluable contribution to both my academic experience and my intellectual growth, for which I will always be grateful.

I would also like to thank my thesis committee members, Dr. Sandip Sen, Dr. Peter van Beek, Dr. Urs Hengartner, and Dr. Theophanis C. Stratopoulos, for their valuable and insightful comments, as well as the help they provided during my studies. I would especially like to thank Dr. Sen, who flew to Waterloo just to attend my thesis defense. I am really happy and honored that I had each one of them on my committee.

In addition, I would like to thank my colleagues and labmates in both the AI and Shoshin Lab for all the fun moments and great discussions we had over the years. Thank you as well to the teammates on my basketball and soccer teams, the Conquerors, the Darkwingers, and the Marxmen, for all the fun games we played.

It is really important when you move to a new place to have someone who already lives there to help you. In my case, I had quite a few people who helped me, my first roommates Jessica Fuchs and Ransy Thorsdottir (go Unit 64 go!), my first officemate, Hao Chen or better now Dr. Hao Chen, and Mr. Christos Diamantis. Their support, especially during my adjustment period, is invaluable.

I would also like to thank my good friends here in Canada, including Spyros Angelopoulos, Ilias Kotsireas, Olga and Tim Baysal, Sharad Birmiwal, Jennifer Jen, and Simina Branzei for their friendship, the great discussions we had, and the enjoyable trips we took. I look forward to traveling around Greece with them when they come to visit me.

Being thousands of kilometers away from home is not easy. However, the support I received from my friends in Greece made my time here a lot easier. I thank them all, especially my best friends, Dimitris Patounis and Eleni Ahtsioglou.

I would like to express my gratitude to Dr. Georgios Lalasidis, Mrs. Hatzisavva , and the Patounis family, especially Mrs. Vasso Patouni. All of them went above and beyond in order to help me keep my job as a teacher in Greece while I pursued my PhD studies. I really appreciate all their help and will always be grateful to them.

Since grade 1^{st}, I had the pleasure and luck of being the student of a number of wonderful teachers and professors who have all helped me become the person I am today. I thank them all and wish them the best. In particular, I would like to thank my Master's thesis advisors, Prof. Leonidas Palios and Prof. Georgios Akrivis, for being such wonderful role models.

My family has always been there for me: my Mom, Regina, my Dad, Kostas, my sister, Despoina, my brother, Theodoris, and of course my grand-mother, Despoina. I am really grateful to them for that. I would especially like to thank my sister for her support and for making sure that I was never short of Greek chocolate and pastries (although I know that my parents paid for them!), and my Mom, who not only managed to raise three children pretty much by herself (since my Dad's job often forced him to be away for extended periods of time), but also made us all useful members of the community.

Last, but certainly not least, I would like to express my deep and sincere gratitude to my good friend Richard Jang (soon to be Dr. Richard Jang), whose friendship and support helped me the most in finishing this thesis. Although I do not look Chinese, my heart feels both Greek and Chinese. Richard has been a great friend. He was always there to share both the good and bad moments. I wish him the best, both for his personal and professional life.

Dedication

I would like to dedicate this thesis to Richard Jang, Kate Larson, and my family. Without them it would not have been possible. This thesis is also dedicated to the memory of all the young people who did not have the opportunity to make their dreams come true as I did. I wish I could change the world and give them the opportunity I had.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the last decade the variety and the number of the services users can access through the Internet have significantly increased. We have reached the point where users can enjoy services that include searching to locate specialized and personalized information, obtaining recommendations, participating in online auctions where users can act both as sellers and bidders, exchanging information, and participating in discussion forums where users can be both questioners and answerers and many more.

Due to the Internet's global nature, millions of users are able to access these services. Since several of them are offered by users themselves, the total number of the services available is rapidly increasing. This enormous amount of information and services is leading people to seek more efficient ways to access them. One way that appears to be dominating is the organization of communities of like-minded individuals who share common goals and interests. By doing so, users try to restrict the services and information they consider, before they decide which ones they will access.

In this thesis we focus on the class of communities in which the services offered are based significantly on peer contributions. We will refer to them as *peer-based communities*. Examples of these communities include, but are not limited to, electronic auctions (e.g. ebay), P2P file sharing networks, recommendation systems, online forums, wikis and many more. One critical observation is that the performance and the popularity of such communities depend both on the number of their members and on the quantity and on the quality of the contribution that each member provides. It is therefore important for communities to make careful decisions when accepting a new member. Furthermore, it is likewise beneficial to encourage users to be good contributors, in their communities.

Since we consider electronic communities, each user is represented by an intelligent agent (i.e., software that aims to perform tasks on the behalf of users to meet their preferences), and thus for the rest of the thesis we will simply refer to the agents that represent the users. An intelligent agent is essentially a piece of software that can act on the behalf of a user.

The ideal situation for a community is when it consists of honest agents that contribute information and services relevant to the interests of other participant agents and to the interests of the community. For example, in an electronic auction community, the desirable situation would be when sellers provide goods at a quality and price that buyers would be interested in purchasing. However, the ideal situation is far from reality. A number of different categories of agents can act maliciously, either deliberately or unintentionally. In particular, these categories are:

- *free riders* which are agents that take advantage of the services provided but never contribute[1],
- *whitewashers* which are agents that keep changing their identity in order to continuously get the benefits of newcomers by resetting their history,
- *malicious agents* which are agents that aim to sabotage the community,
- *selfish agents* which are agents that aim to temporarily gain as much as possible without having any concern that this might decrease both the performance of the system and its long-term benefits,
- *careless agents* which are agents that unintentionally overload the environment with unnecessary or bad quality services or information.

The communities we consider in this thesis are multiagent systems and thus they consist of multiple agents that interact with each other. In these communities there are two main ways to motivate agents to behave in a desirable manner. The first one is based on the assumption that the agents can see their long term benefits. More specifically, the agents can reason that if all the agents in the system contribute good quality of information then in the future they will all be able to utilize better services. However, given that agents represent users and thus act based on the users' strategies, a scenario in which all the agents would be able to see the long term benefits they might acquire can be considered unrealistic. In order to deal with this problem, a second way is through the development of an incentive-based model that essentially determines the benefits an agent receives based on its behaviour. In particular, from our perspective, the idea of using incentives is to "motivate" agents to act in a desirable manner by "threatening" to directly or indirectly limit their access to the system's privileges. Usually, the desirable manner refers to honest behaviour. For example, honest revelation of information, honest feedback etc.

[1] The existence of free riders results in a problem that is commonly known as *the tragedy of the commons* [21].

In this thesis we consider a novel approach of providing incentives to agents to be good participants. This approach is based on the observation that people and even animals tend to behave better when they feel that their behaviour is being monitored. This is commonly known as the *Hawthorne Effect* [18]. Similar to this, we consider an approach where agents know that their behaviour inside a community will be observed and will be shared with the communities they might be interested in joining in the future. Since agents can migrate between, or be members of, multiple communities, how an agent behaves in one community is of interest to others. This is particularly true in situations where communities can decide whether or not to allow a particular agent to join. When an agent attempts to join a new community, it should be possible for this community to acquire information about the agent's past history, as part of its reasoning about the potential value of the agent. In cases where communities have limited ability to accept new members, those agents that appear to be more reliable and a better match will be more likely to be accepted into the community. At the same time, agents will realize that their previous *evaluation* is going to be considered whenever they choose to join new communities. As a result, they may be inclined to participate well and in a trustworthy manner, no matter which community they inhabit.

Summarizing, we argue that having communities communicate regarding the behaviour of their agents will have the following important effects:

- good agents are identified, so that they can enjoy benefits right away,
- the average contribution of agents in the community may increase, since agents will be motivated to contribute; otherwise, this will have a negative impact on their evaluation,
- the services and information provided will be more useful, since the communities will end up having agents that better match their profile.

By allowing communities to exchange their experiences about their agents they can: i) exploit the information that each community accumulates while at the same time communities put pressure on agents by informing them that any misbehaviour will be spread to the communities they might wish to join in the future, and ii) prevent agents from overstretching themselves among many communities, since this may lead them to provide very limited services to each of these communities due to their limited resources.

Exchanging evaluation ratings cannot be effective without the existence of an accurate model for evaluating the behaviour of the agents in communities. The framework we propose is as follows. Within each community, an agent earns a certain evaluation rating, based on the value of its participation within the community and the quality of its interactions with other members. Consequently, the way the behaviour of each agent is modeled is crucial. The evaluation of an agent can be either the reputation of the agent or the trustworthiness of the agent.

Although, trust and reputation have been used by a number of researchers interchangeably [25, 77] in this thesis, we argue that trust should be distinguished from reputation and we focus on providing a novel trust model for evaluating an agent since trust is a more objective and comprehensive metric of the behaviour of an agent. In particular, by *trust* we refer to the degree an agent honored its promise and by *reputation* we refer to the degree of satisfaction (i.e., contribution to the maximization of their utility) the agent brought to the other agents with which it has interacted. Although trust and reputation can be related to each other, one does not necessarily imply the other. For example, an agent might have a high reputation but can be proved untrustworthy in an interaction with another agent.

Another limitation with using reputation ratings for the case of communities is that the values provided are highly related to the particular needs of the community. For example, a desirable agent might have low reputation inside a community because although it is willing to contribute, there is no current interest in the services it offers or simply the services it offers do not much with the needs of the community. On the other hand a malicious agent can temporarily create a good reputation and then become deceptive in the future, once it has been accepted in another community. For the aforementioned reasons, in this thesis by evaluation rating we will refer to trustworthiness rating.

Summarizing, the problems we address in this thesis are the following:

1. Exchange of information between communities:
   (a) *Advisor Selection Problem*: Select the set of communities to exchange information.
   (b) *Payment Decision Problem*: Determine the compensation a community should receive with respect to the quality of the information it provided.
2. *Trust Modeling Problem*: Modeling of trust in peer-based communities.

1.1 Exchange of Information Between Communities

Several researchers have proposed the exchange of ratings of agents between peers in social networks in the context of modeling the reputation and trustworthiness of agents [31, 46, 76, 91, 92]. This is especially useful in settings such as e-marketplaces, where a buying agent might have little experience with a selling agent and may choose to ask other buyers for advice. In our work, we examine a similar but distinct problem: how to promote honest exchange of information about the evaluation of agents, between communities.

Communities are primarily self-interested and thus they have strong incentives to play strategically and occasionally misreport the evaluation of their agents. For example, a community C_A could be reluctant to provide a truthful evaluation for one of its very good agents to another community C_B. This is due to the fact that if the community C_B accepts the agent then C_A might have to share the resources that the agent contributes with C_B. Given that the agent's resources can be limited, this can result in a decrease of the agent's contribution to C_A. Furthermore, a community might lie about an agent with a poor contribution in an effort to get rid of them.

We would like to note that in our model, we consider that the *honest* communities can accurately reason about the behaviour of their agents and that their agents do not change their behaviour from one community to another. In addition we consider that a community which does not (strategically or not) or cannot reason about the value of its agents is *dishonest*. In other words by *honest* we refer to a community that provides valuable information while by *dishonest* we refer to a community that provides information of no significant value. This is because each community is self-interested and thus it would like to collaborate with communities that provide information that it can use. Even if a community C_A was truthful about an agent, and the agent proved to honestly behave differently, the information that C_A provided is not valuable to the recipient community, and thus it is rational for the recipient community to decrease the probability of asking that community in the future. Similar to this is the case where the inaccurate information was due to the lack of a good evaluation model. If a community is honest but uses an inaccurate evaluation model to reason about its agents then, again, the information that it provides to other communities might not be helpful.

Our aim is to design a framework that enables communities to truthfully share information about their agents. The key idea of our research is to promote truthfulness both amongst agents and communities. Agents should be inclined to be good citizens within their communities because their reputation will be shared; communities need to be inclined to honestly share the reputation ratings of their agents in order to be able to benefit from the information provided by the other communities in the system. For this reason we propose a two phase incentive mechanism. In the first phase our approach determines the set of communities that will exchange information. In particular, it determines which communities the recipient community will consult in order to acquire information for an agent. We refer to these communities as the *advisor communities*. In the second phase we propose a payment function that determines the compensation each advisor should receive based on the quality of the report it provided.[2]

[2] The quality of a report is discussed in details in Chapter 5.

Selecting Communities to Consult

For addressing the problem of selecting which communities to consult we propose the use of a graph-based heuristic which exploits the consistency among the advice of candidate advisors and reduces the number of communities that are queried each time a community seeks information about a prospective agent. Reducing the number of communities that are queried is very important since each time a community requests information from another community it has to provide some resources while also it can provide strong incentives to the communities to provide truthful reports. However, the selection should be determined carefully in order for all the communities to be inclined to provide truthful reports.

We argue that exploiting consistency among good advisors will further promote honest behaviour since fewer communities will be consulted and thus resources will be restricted to only the most reliable sources. This will create a more competitive environment between the advisor communities, which will be inclined to be truthful in order to increase the probability of being asked in the future. Moreover by using a heuristic that exploits consistency we can prevent the communities from playing strategically when deciding when to tell the truth and when to lie. Note that simply selecting the communities with the highest probability of telling the truth is not always sufficient, since it might be the case that although a community C_A can have a probability of telling the truth a little bit below the probability of another community C_B, the community C_A might be a better choice than C_B. One reason that C_A might be better is if the transactions based on which the latter probability is determined are more recent than the ones that determine the probability that C_B is telling the truth.

Determining Payments

The goal of our payment function is to determine the compensation each community should receive based on the quality of the report it provided in a context where the exchanged good is information and, in particular, evaluation ratings. The two main issues we are interested in addressing are:

1. how a community can be motivated to truthfully report its ratings and
2. how we can value an evaluation rating an advisor community provides in order to compensate the community with a fair payment.

As will be seen, we value the rating a community provided through a function that we refer to as the *Importance Function*. We will refer to this function as the *i-Function* and to its value as the *i-Score*. We set the *i-Function* to be maximized only when the rating is a truthful rating (for 1.) and introduce a set of properties it should follow in order to promote honesty and fairness (for 2.), thus, providing an effective proposal for the exchange of evaluation information between communities. The compensation each community will receive is monotonically increasing with respect to the *i-Score* it was assigned to the rating it provided.

A novel and key distinction that we make is between what we refer to as a *good* or a *poor* contributor. These are labels that correspond to the interpretation of a rating since different communities might use different evaluation models. More specifically, the information that is shared consists of both an evaluation rating that can either be a trust or reputation rating and a type (*good* or *poor*) which essentially is the interpretation of the latter rating. The community receiving the ratings from each community explicitly evaluates the accuracy and helpfulness of the information the advisor communities provided. As will be shown, information that leads to the correct decision about accepting or rejecting an agent leads to a higher *i-Score* (and thus to more lucrative payments) to the advisors, thus promoting both honest reporting and fair payments.

1.2 Modeling Trust

Similar to Josang [28] and Wang and Vassileva [81] in our work we consider *trust* and *reputation* as two distinct terms. We first introduce an arrangement where agents are required to declare a promise for their contribution to a community. We then leverage that promise as part of the trust modeling as follows. By *trust* we refer to the degree an agent honored its promise.

In this thesis, we focus on developing a novel trust model for settings in which important decisions are made based on the promised contributions of the participants. In particular, we present this as the basis on which a community can judge the trustworthiness of its agents. We note that our model is intended for communities where an agent's value is dependent in part on the extent to which it contributes its fair share to the community (hence the focus on an agent's promise). We consider trust as the *degree an agent honored its promises* and we propose a trust model that has the following properties: i) promotes a desirable behaviour which is the truthful revelation of the agents' anticipated contributions and at the same time contributes to the community, and ii) is able to cope with some inaccurate promises due to lack of information. We show that these properties ensure that an agent maximizes its trust score by both delivering what it promised and by promising what it can deliver and we provide functions which embody these principles. Finally, we discuss how our model can be adapted to settings where, due to changing circumstances, an agent may be unsure when they make a promise that they will be able to deliver as desired and we provide an empirical evaluation.

1.3 Structure of the Thesis

In Chapter 2 we provide an overview of current approaches that address problems similar to ours or that have provided useful tools for our proposed solution. In Chapter 3 we present the overall model and the formal definition of the three problems we address: the *Advisor Selection Problem*, the *Payment Decision Problem* and the *Trust Modeling Problem*. In Chapter 4, we provide our suggested solution for the *Advisor Selection Problem* and we show that exploiting consistency between advice provides a significantly effective tool in incentive mechanism design. In Chapter 5 we present our proposed solution for the *Payment Decision Problem* by extending a well known family of functions called *Scoring Rules* [67]. In Chapter 6 we present our solution for the *Trust Modeling Problem* by considering the degree to which an agent honored its promises. In Chapter 7 we discuss how our solutions differ from current proposed approaches and why they provide important contributions to the areas of information sharing and trust modeling. In Chapter 8 we outline our key contributions, the current open problems, and possible directions for their solution.

1.4 Thesis' Statement

The number and the population of electronic communities in which the services offered depend on the contributions of their members is rapidly increasing. It is therefore important to prevent, within these communities, the existence of malicious participants and those which make poor contributions.

In this thesis we show that by providing the communities with a mechanism that allows them to truthfully exchange information with other communities with respect to the trustworthiness of their participants, the communities

- can deal with malicious participants,
- encourage poor contributors to increase their contribution, and
- provide incentives to good contributors to maintain or further improve their services.

The truthful participation of the communities in the mechanism and the adaptation of an accurate model that evaluates the trustworthiness of each participant are crucial for the success of the mechanism.

In particular, we propose:

- a two phase mechanism that promotes truthfulness by exploiting the consistency of the information different communities provide, and

- the modeling of the trustworthiness of the participants in a community based on the extent they honor their promises.

Chapter 2

Related Research

Several subareas within the area of Multiagent Systems in Artificial Intelligence serve as the background for our problem and within these areas, certain models developed by other researchers were particularly useful to examine. More specifically, these areas include *Trust & Reputation* and *Incentive Mechanisms* (a subarea of Mechanism Design). Furthermore, the area of *Statistics* and in particular *Scoring Rules* and the area of *Graph Theory* have provided us with valuable tools to develop our solutions. Below we give a brief summary of each relevant topic area, along with a description of some approaches within those areas that either provide interesting solutions or that we found particularly important.

2.1 Trust and Reputation

Several multiagent systems researchers have proposed frameworks for modeling the trustworthiness and the reputation of agents, towards effective selection of systems that include evaluating potential sellers in e-marketplaces [77, 76], monitoring the nodes in sensor networks [47], relaying messages [41], handling pollution with inauthentic files in peer to peer networks [34], task allocation [72], coalition formation [66, 71], electronic supply chains [70] and many more.

A valuable survey on trust for multiagent systems was presented in 2004 by **Ramchurn et al.** [54]. In particular, the authors presented the state-of-the-art trust models along with an analysis of their strengths and weaknesses and suggested an interesting list of future research directions. In 2007, **Josang et al.** [29] presented a survey on trust and reputation for online service provision. The authors analyzed trust and reputation systems and their differences, while they also described in which cases these two systems overlap. In particular, they discussed the different types of reputation systems, the different architectures, and also the variety of ways that both trust and reputation have been interpreted by researchers. Furthermore, they provided a very interesting observation that helps to explain why most commercial systems use very simplistic approaches, which is: *from a business perspective having a reputation system that it is not robust can be desirable if it generally gives a positive bias.* In other words, giving the clients the feeling that they can have some kind of information about the behaviour (reputation or trustworthiness) of the service provider makes them feel more confident regarding the reliability and the trustworthiness of the whole system.

There are a number of different ways to measure reputation and trust from the simplest ones like models based on summations or on average ratings [1, 2], to more sophisticated and popular probabilistic approaches [10, 25, 59, 58, 64, 76], to discrete models (e.g. in amazon.com [1]) buyers can rate the book reviews as *helpful* or *not helpful*), and finally to flow models, like Google's [49] approach, which measures the reputation of pages based mainly on the number of links that other pages might have to them, or models based on social networks [53]. In the next paragraphs we present several systems that describe each of the above models.

Very simple systems for measuring the reputation of an agent are the ones that ebay [2, 62], an auction based e-marketplace, and Amazon.com [1], a book recommendation and marketplace, use. More specifically, ebay uses a summation model in which at the end of each transaction both the seller and the buyer can rate each other. The rating scale they can use is 1 for a satisfying transaction, 0 for neutral and -1 for indicating a low quality transaction. The reputation of each participant is counted by summing the number of the positive ratings and then deducting the number of negative ratings it received. In particular:

$$rating = \sum_{\forall r_i^+ \in R_{pos}} r_i^+ - \sum_{\forall r_i^- \in R_{neg}} r_i^-$$

In order to avoid ballot box stuffing e-bay charges a minimum fee to any seller interested in selling a good.[1] Even though the above idea can lead to reducing the number of buyers who try to deceive the system by completing fake transactions, the ballot stuffing problem cannot be fully eliminated. A malicious agent can still deceive the system by spending money setting up fake transactions for very low priced items in order to build up its reputation with the anticipation that later on it can set up a fake auction about a very high priced item. This is due to the fact that ebay considers that the importance of the ratings for a transaction are independent from the value of the item that was sold. This means that a seller that sells products in the range of few dollars, with 10 positive ratings and 0 negative ratings will have the same reputation as a seller who sells goods in the range of several thousands dollars and has exactly the same ratings. Intuitively, someone could argue that it is more likely that the second seller is more reliable than the first one.

[1] Ballot box stuffing is the situation that occurs when agents can vote positively multiple times without these corresponding to real transactions [14, 15].

A more loose approach has been adopted by Amazon.com [1, 29]. In Amazon.com, users can add a review for a book and/or a rating for the book in the scale of 1 to 5. Other users can read these reviews and rate them as *helpful* or *not helpful*. Amazon.com uses the number of helpful votes each reviewer has received in order to determine the ranking of the reviewers. In an effort to prevent ballot box stuffing Amazon.com considers a number of other parameters, which it does not reveal in order to prevent malicious users from manipulating the ranking mechanism. Amazon.com also allows one review vote per registered cookie. Unfortunately, as it is also stated in [29], Amazon.com's efforts are not sufficient for resolving the problem of ballot box stuffing, since resetting a cookie or changing computers can allow a user to provide multiple votes for a particular book or review. Apparently, badmouthing can also occur since it is impossible to detect whether a user provided honest feedback regarding a book/review or if he/she is just acting maliciously.

In 2002, **Josang and Ismail** suggested the Beta Reputation System [25]. This work introduces a new mechanism for computing the reputation of an entity based on the history of its behaviour and has been used as a stepping stone for other approaches. This is for the context where one agent asks other agents for reputation ratings, which can be either positive or negative. The mechanism is heavily based on the beta probability function, which is a function that can be used for binary events. In particular, the authors consider the gamma function Γ:

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

with the following restrictions: $p \neq 0$ if $\alpha < 1$, and $p \neq 1$ if $\beta < 1$. As the authors note using the above function appears to have minimal significance. The important aspect proves to be the expectation value of the beta distribution, which is:

$$E(p) = \frac{\alpha}{\alpha+\beta} \tag{2.1}$$

In the reputation system, α represents the number of positive feedbacks and β represents the number of negative feedbacks. More specifically, in order to reason regarding the probability of an agent to be truthful or not, the equation in (2.1) will take the following form:

$$E(p) = \frac{r+1}{r+s+2}$$

which actually comes by replacing the α and β in (2.1) with $r+1$ and $s+1$ respectively. r denotes the number of positive feedbacks that the entity received and s denotes the number of negative feedbacks. In order to extend the mechanism to accommodate cases where the significance of the feedback is based on the trustworthiness of the agent that provides it, the authors incorporate a discounting factor.

A more concrete effort was made to consider the problems that emerge when an agent changes its behaviour. Predicting the agent's behaviour based on its history is a very difficult task since it must successfully detect and handle cases in which the agent may have changed its behaviour. In order to accommodate such situations, the authors propose a way to calculate the number of positive and negative feedbacks based on the sequence of each rating. In particular, they consider a number of agents that provided a sequence Q containing n feedback tuples $(r^Q_{T,i}, s^Q_{T,i})$. In order to calculate the total number of positive and negative feedbacks, $r^Q_{T,\lambda}$, and $s^Q_{T,\lambda}$ respectively, they propose the following formula:

$$r^Q_{T,\lambda} = \sum_{i=1}^{n} r^Q_{T,i}\lambda^{(n-1)} \qquad \text{and} \qquad s^Q_T = \sum_{i=1}^{n} s^Q_{T,i}\lambda^{(n-1)}$$

where $0 \leq \lambda \leq 1$, and $r^Q_{T,i}$ is newer than $r^Q_{T,i+1}$, and $s^Q_{T,i}$ is newer than $s^Q_{T,i+1}$, $\forall 1 \leq i < n$. When $\lambda = 1$ then all the feedback is equally significant, and when $\lambda = 0$ the authors will consider only the last feedback.

Finally, the authors consider the problem of requesting users to provide feedback as a pair. Users will be more willing to just provide a single value than to provide a pair. For this reason they define a normalization weight w which is equal to the sum of r and s, w=r+s, and request that the users provide this value instead of values for both r and s. Concluding, the beta reputation system presented in [25] encapsulates interesting mathematical concepts of use in evaluating reputation ratings provided by other agents, while in addition has inspired a number of researchers in developing mainly reputation models.

Also in 2002, **Sen and Sajja** [72] proposed a boolean reputation-based trust model to address the problem of agents (*user agents*) selecting agents (*processor agents*) to process tasks. The authors provide a reinforcement learning based technique to estimate the performance of a processor agent based on its past interactions. When a user agent requests information regarding a processor agent from a set of agents S, it receives a boolean answer from each agent. The latter answer can take the values either *high performance* or *low performance*. In order for a user agent i to decide whether a processor agent j is a *high performer* or *low performer* it computes an estimation e^t_{ij}. If $e^t_{ij} < 0.5$ agent i reasons that agent j is a low-performer and high-performer otherwise. A reinforcement learning technique is used to calculate the estimation e^t_{ij} based on whether the agent i got to interact with agent j or agent i observed the interaction of j with another user agent. In particular, if i has interacted with j we have:

$$e^t_{ij} = (1 - a_i)e^{t-1}_{ij} + a_i r_{t-1} \tag{2.2}$$

while if i has observed agent j:

$$e_{ij}^{t} = (1 - a_0)e_{ij}^{t-1} + a_0 r_{t-1} \tag{2.3}$$

where a_i and a_0 are the interaction and observation specific learning rates, respectively, and r_{t-1} is the performance received or observed in time $t-1$.

If the majority indicates that a processor agent provides high performance services the agent is added in the *preferred list*, while otherwise it is added in the *uncertainty list*. If the preferred list is empty then a processor agent will be randomly selected from the uncertainty list, while otherwise it will be selected randomly from the preferred list. The next question the authors address is: What is the minimum number q of agents that should be queried? The minimum number of agents to query is defined as the minimum value of the parameter q that satisfies the following inequality:

$$\sum_{i=max(\lceil \frac{q}{2} \rceil, \lceil \frac{q}{2} \rceil + 1)}^{P} \frac{\binom{N-l}{i}\binom{l}{q-i}}{\binom{N}{q}} \geq g \tag{2.4}$$

where P is the number of processor agents, l is the number of liars (where $l \leq \frac{N}{2}$), N is the number of user agents, and g is the minimum desired probability of selecting a high-performer processor agent. The summation represents the probability that at least a majority of the q selected agents are non-liars. The authors present a set of experiments that demonstrate the robustness of their proposed model with respect to the varying of the number of liars that might exist.

In 2003, **Kamvar et al.** [34] suggested a solution for the problem of the pollution of a P2P network with inauthentic files, which results in reducing the performance and reliability of the network. Their main assumption rests on the existence of a significant number of pre-trusted peers. The authors presented a distributed algorithm based on Eigenvectors according to which each peer calculates the global trust for each other peer in the network based on the local trust values that it stores. The latter trust value is derived from its interactions with other peers. This follows the same philosophy that ebay uses with the difference that the computation of the global trust takes place in a distributed manner. More specifically, it considers the number of satisfactory, $sat(i,j)$ and unsatisfactory, $unsat(i,j)$, transactions a peer i had with peer j:

$$s_{i,j} = sat(i,j) - unsat(i,j)$$

The next step is to calculate the normalized local trust value c_{ij}:

$$c_{ij} = \frac{max(s_{ij}, 0)}{\sum_j max(s_{ij}, 0)}$$

The following step is to calculate the trust t_{ik} that peer places in peer k by asking its friends:

$$t_{ik} = \sum_j c_{ij} c_{jk} \tag{2.5}$$

If we consider C to be the matrix $[c_{ij}]$ and $\vec{t_i}$ to be the vector containing the values t_{ik}, then the equation (2.5) can be written as $\vec{t_i}{=}C^T\vec{c_i}$. In order to have a complete view the peer i has to ask the friends of its friends and so on up to depth n. At this point the authors argue that if n is large enough then the trust vectors that each peer i calculated for all the other peers will converge, and more specifically they will converge to the left principal eigenvector of C.[2]

The above algorithm puts more weight on the opinion of peers that the peer trusts, while it considers a probabilistic approach for giving chances to newcomers that have not yet gained a reputation. Even though the authors provide some interesting observations, it is still not clear to what extent their mechanism is scalable and distributed. In addition, information related to the size of the inauthentic files a peer downloads was not taken into consideration (e.g. downloading a big unauthentic file can be more annoying than downloading a small one, since in the first case the agent has to spend more resources). Finally, the authors point out the problem of the existence of malicious collectives (coalitions) while they put some effort into considering the problem of free riders and whitewashers.

In 2004, **Tran and Cohen** [77] presented a reinforcement learning based mechanism for updating the reputation of selling agents in e-marketplaces. In particular, the mechanism allows both the selling agents to adjust the price and the quality of their goods and also allows the buyers to adjust their purchasing decisions based on the reputation of the sellers. The reputation of a seller is based on the direct experience of the buyer with the particular seller. A seller s can be either characterized by a buyer b as *reputable* or *disreputable* or *neither reputable or disreputable*. The category in which a seller belongs is computed based on its reputation. More specifically, the authors propose that each buyer considers two thresholds, θ and Θ, where θ and Θ are the buyer's reputation and disreputation thresholds respectively. These thresholds are defined by each buyer to depict the level of tolerance the buyer has with respect to the seller's reputation. For example, a buyer b_0 can characterize a seller that has reputation equal to 0.6 as *reputable* while in contrast another buyer b_1 might characterize the same seller as *neither reputable or disreputable*. Each time a transaction is completed the reputation of the selling agent is updated according to the expected value of a good (a function of quality and price). The better the value of the good the buyer received with respect to the value it requested, the more the reputation of the seller will be increased.

[2] For a detailed technical analysis please refer to [34].

In general a buyer is interested in doing business with the sellers that are *reputable* rather than with the sellers that are *disreputable* or *neither reputable or disreputable.* The key point is that the buyer will not necessarily select the most reputable agent. In fact the selection is based on a function f which essentially depicts the trustworthiness of the seller. In particular, consider a buyer b who is interested in purchasing a good g at a price p. This buyer will choose the seller s^* that maximizes a function f such that:

$$s^* = \arg\max_{s \in S_r^s} f^b(g, p, s)$$

where f reflects the expected value of the good and is updated each time a transaction between the seller and the buyer has been completed. More specifically:

$$f^b(g, p, s^*) = f^b(g, p, s^*) + \alpha * (f^b(g, p, s^*) - \upsilon^b(g, p, s^*))$$

where α, $0 \leq \alpha \leq 1$, is a learning rate and $\upsilon^b(g, p, s^*)$ represents the true product value of the good, determined by the buyer upon observing the good, after purchase.

After each transaction the reputation of a seller is also updated. Very briefly, the reputation, $r^b(s^*)$, of a seller s^* will be updated based on whether it provided a good whose quality was satisfactory or not, with respect to the price the buyer paid. In case the buyer is satisfied then the following formula is used:

$$r^b(s^*) = \begin{cases} r^b(s^*) + \mu(1 - r^b(s^*)) & \text{if } r^b(s^*) \geq 0 \\ r^b(s^*) + \mu(1 + r^b(s^*)) & \text{if } r^b(s^*) < 0 \end{cases}$$

where μ is a positive and is called *cooperative factor.* Unlike ebay [2], where the price of the merchandize does not influence the importance of a rating, in this approach the factor μ considers the value of the good. A detailed description on how to compute μ and the formula for updating the reputation in case the buyer was not satisfied with the quality of the good can be found in [77].

Moreover, in order to allow newcomer sellers to start business, the authors consider that each buyer with a small probability chooses to "explore" the marketplace by selecting a seller that has reputation below the reputation threshold Θ. Even though this will increase the chance that a buyer will seek to purchase items from a newcomer, is not clear if this increment will be significant enough to decrease the inactive time that a trustworthy seller has to spend before accumulating reputation that will allow it to be competitive. For example, consider a seller that is interested in selling tickets for tomorrow's basketball game, if i) it has to wait for more than a day to accumulate reputation that will make it competitive with respect to the other agents that sell basketball tickets or ii) if this seller was not lucky enough to be chosen through the random selection procedure, then it will not be able to sell the tickets.

This is exactly the problem our approach aims to eliminate through the exploitation of an agent's good behaviour in previous communities. Very briefly, if the seller s_0 had built a good reputation and was a trustworthy agent in the communities it belonged to so far, then it would be desirable for this agent to utilize this good behaviour in the community that uses a mechanism similar to the one presented in [77]. Concluding, the key idea of the proposed approach is that the reputation of an agent should not only consider the number of contributions the agent made to a community but also the value/quality of the items that he/she contributed.

In 2004, **Yu et al.** [87] addressed the problem of measuring trust through reputation in large scale peer to peer systems when direct experience is not sufficient. If the combination of direct experience (if any) and peer information (if necessary) exceeds a threshold the peer is considered trustworthy. Given that peer to peer networks tend to be large, the use of a centralized approach can be significantly inefficient. In order to overcome this problem, the authors propose a distributed reputation system that aims to detect malicious and unreliable peers. Similarly to the Beta Reputation System [25], a peer P_i which is trying to discover the trustworthiness of another peer P_j can either use its own opinion that is derived from its personal experience of interacting in the past with P_j or a combination of personal experience and the reputation of the peer P_j provided by other peers.

Unlike other proposed approaches, which mainly consider binary ratings (e.g. "satisfied" or "not satisfied"), the authors consider that the ratings should encapsulate the quality of the provided service. For this reason, they consider that the value of a rating is inside the interval [0,1].

A peer P_i can either use a simple averaging method or it can use an exponential method to measure the reputation of a peer P_j based on its personal interactions.

- The Averaging Method:

$$R(P_i, P_j) = \begin{cases} \sum_{k=1}^{h} s_{ij}^{k}/h & h \neq 0 \\ 0 & h = 0 \end{cases}$$

  where h denotes the number of the last interactions the peer P_i considers, and s_{ij}^k, $0 \leq s_{ij}^k \leq 1$, is the rating that peer P_i assigned to peer P_j at their k-th last interaction.

- The Exponential method[3]:

$$R(P_i, P_j) = \begin{cases} \gamma[s_{ij}^{h} + ... + (1-\gamma)^h s_{ij}^{1}] & h \neq 0 \\ 0 & h = 0 \end{cases}$$

[3] Note: We are not sure whether this formula, as presented in [87], is correct.

where γ, $0 < \gamma < 1$, represents a weight that determines the influence of ratings with respect to time. The bigger the value of γ the faster the past ratings are forgotten.

If the peer P_i is not confident regarding its personal experience with peer P_j, it will seek information from other peers that had personal interactions with the peer P_j. Then the aggregated rating for peer P_j is calculated as follows:

$$T(P_i, P_j) = \begin{cases} \eta R(P_i, P_j) + (1-\eta)\Upsilon & L \neq 0 \\ 0.5 & L = 0 \end{cases}$$

where $\eta = h/H$, L is the number of witnesses found by peer P_i, and

$$\Upsilon = \begin{cases} \sum_{k=1}^{L} w_k \cdot R(W_k, P_j) & L \neq 0 \\ 0.5 & L = 0 \end{cases}$$

where $R(W_k, P_j)$ is witness W_k's local rating for peer P_j, and w_k is the weight of the credibility of witness W_k. By the term "witnesses" the authors refer to the peers from which the peer P_i will request their opinion regarding the peer P_j. If $T(P_i, P_j)$ is above a threshold ω_i the peer P_i will interact with P_j; otherwise, P_i will mark peer P_j as *unreliable.*

The question that arises at this point is: How does the peer P_i decide which peers to consider as witnesses? In order to answer this question the authors considered the following approach: the peer P_i will ask a number of the peers in its neighborhood if they know the peer P_j. In case they do, they will send their personal rating for peer P_j to peer P_i; in case they do not know, they will check if the maximum TTL or query depth has been reached. If not, they will forward the question to a number of their trustworthy neighbors, and so on.

Furthermore, the authors proposed a way of eliminating the influence of noisy ratings (e.g. exaggerated positive or negative ratings) that witness peers might provide. The basic idea is to give more weight to witnesses that provide more accurate information regarding other peers and to penalize (i.e., decrease their weight) peers that provided inaccurate information. In particular, considering a peer P_i that provided a rating s for a service that was served by peer P_j, the weights of the witnesses that provided information to peer P_i about P_j will be updated as follows:

$$w_k = (1 - (1-\beta)|R(W_k, P_j) - s|) * w_k$$

In 2006, **Teacy et al.** [76] presented both a trust model and a reputation model for the case of binary events. The authors present TRAVOS, a model that essentially extends the one presented by Josang and Ismail [25], to cope with the existence of inaccurate ratings. First, a trust model based on the direct experience a truster a_{tr} with a trustee a_{te} is presented. The level of trust based on the interactions since time 1 is determined from the following formula:

$$t_{a_{tr},a_{te}} = \frac{m^{1:t}_{a_{tr},a_{te}} + 1}{m^{1:t}_{a_{tr},a_{te}} + n^{1:t}_{a_{tr},a_{te}} + 2}$$

where $m^{1:t}_{a_{tr},a_{te}}$ is the number of positive interactions and $n^{1:t}_{a_{tr},a_{te}}$ is the number of negative interactions from time 1 to time t, where t is the time of the assessment. Essentially, $t_{a_{tr},a_{te}}$ represents the expected belief of the truster on the trustee.

Clearly, the accuracy of the above value is related to the number of past interactions the truster has with the trustee. The more interactions it has had, the more accurate $t_{a_{tr},a_{te}}$ will be. In order to determine the confidence of the truster with respect to the value $t_{a_{tr},a_{te}}$ a confidence metric $\gamma_{a_{tr},a_{te}}$ is defined. The confidence factor $\gamma_{a_{tr},a_{te}}$ is computed based on the proportion of the probability distribution f that lies in $[t_{a_{tr},a_{te}} - \epsilon, t_{a_{tr},a_{te}} + \epsilon]$ where ϵ is an acceptable error margin. More specifically:

$$\gamma_{a_{tr},a_{te}} = \frac{\int_{t_{a_{tr},a_{te}}-\epsilon}^{t_{a_{tr},a_{te}}+\epsilon} X^{\alpha-1}(1-X)^{\beta-1}dX}{\int_0^1 U^{\alpha-1}(1-U)^{\beta-1}dU}$$

where $\alpha = m^{1:t}_{a_{tr},a_{te}} + 1$ and $\beta = n^{1:t}_{a_{tr},a_{te}} + 1$.

When there is complete lack of direct experience or the confidence factor is low, the authors provide a model that measures the reputation of the agent. The model they propose is similar to one for measuring trust. In the case of reputation, the positive $(m^t_{a_{op},a_{te}})$ and negative $(n^t_{a_{op},a_{te}})$ are defined based on the interactions that each of the raters had with the trustee. By rater we refer to the agent that provides information to the truster. Clearly, the accuracy of the results is influenced from both the truthfulness of each rater and the behaviour of the trustee to each rater. In order to address the above problem, the authors filter the information the raters provided. This is achieved by exploiting the extent that E', which is calculated based on only the experience of the rater with the particular trustee, deviates from E^0, which is calculated based on the interactions that other raters had with the trustee. More specifically:

$$E' = \frac{m^t_{a_{op},a_{te}} + 1}{m^t_{a_{op},a_{te}} + n^t_{a_{op},a_{te}} + 2}$$

and

$$E^0 = \frac{M^t_{a_{op}} + 1}{M^t_{a_{op}} + N^t_{a_{op}} + 2}$$

where $M^t_{a_{op}} = \sum_i m^t_{a_{op},a_i}$ and $N^t_{a_{op}} = \sum_i n^t_{a_{op},a_i}$.

If both E' and E^0 lie in the same interval (known as *bin*) the interaction is considered to be successful; otherwise it is considered to have failed. Based on the successful and the failed interactions the probability of the accuracy of the reported opinion is calculated. This probability is then used to determine the values m_{op} and n_{op} which are used for calculating $t_{a_{op},a_{te}}$, since:

$$t_{a_{op},a_{te}} = \frac{m_{op} + 1}{m_{op} + n_{op} + 2}$$

The above procedure can deal with inaccurate ratings when measuring the reputability of an agent since it does not allow big deviations to influence the reputation rating. Although this brings a certain level of stability in the model, in a number of cases it may allow self-interested agents to play strategically.

In 2006, **Regan et al.** [59] proposed a Bayesian based approach for interpreting the sellers' evaluations in e-marketplaces called *BLADE* (*B*ayesian *L*earning to *A*dapt to *D*eception in *E*-marketplaces). As the authors argue, their model is a generalization of the BRS [25] and TRAVOS [76] with the following two improvements: i) goes beyond a single reputation value and models a set of seller properties, and ii) allows the buyer to re-interpret any ratings that are not a direct mapping of the seller properties. More specifically, the authors consider an e-marketplace where buyers can ask the advice of other agents (advisors) regarding the reputation and trustworthiness of sellers. Since each advisor has its own way of evaluating a seller, simply knowing the value of a rating cannot provide accurate information regarding the seller. For example, an advisor can be in general very lenient while at the same time another advisor might be very strict. Consequently, a rating with a value equal to 0.6 might have totally different interpretations depending on who was the advisor that provided it. Furthermore, there are also cases in which an advisor might deliberately provide inaccurate information about a seller while in addition both sellers and advisors might change their behaviour over time.

In order to deal with the above problems the authors propose a probabilistic reasoning technique which aims to learn the way advisors evaluate sellers, deals with the problem of misleading evaluations and finally is flexible enough to adapt to the changes in the behaviour of both sellers and advisors. More specifically, the proposed mechanism is based on the use of a Bayesian Network which exploits the set of features that each seller exhibits and the evaluation of each advisor. The conclusion of this work is that the uncertainty of the adequacy of the information an agent has regarding the reputation of the other agents can significantly influence the "correctness" of its decisions, while the accuracy of an agent's ratings is dependent on his/her personal style of evaluation.

Although reputation is an important factor for measuring the trustworthiness of an agent or a user [61], it does not always encapsulate significant information for identifying trustworthy parties. An interesting analysis that provides insights of the other components that can be used in a trust management system is presented by **Pourshahid and Tran** in 2007 [52]. Furthermore, the proposed approach illustrates that the special characteristics of each environment constitute a crucial factor for determining the components that should be taken into consideration (e.g. time might be crucial for an environment that provides hockey tickets; the exact condition of collective books is crucial while in other cases as long as the book is in a decent state and the buyer did not have to pay too much then she might be satisfied) and demonstrates the fact that people have different methods of measuring trust. Moreover, the authors agreed with the belief that many researchers express that trust is a combination of logical, emotional and physiological factors that contribute either directly or indirectly to the decision making process.

2.2 Mechanism Design: Incentive Mechanisms

Assigning a reputation or trust rating that depicts the behaviour of an agent can certainly motivate agents to improve their interactions, however it cannot eliminate strategic behaviour. In order to address this problem the area of *Incentive Mechanisms* has emerged. The main idea is to set the rules by which agents will interact, in order to maximize some global utility through the operation of individually rational agents. For the case of incentive mechanisms we seek rules that provide incentives for agents to have truthful behaviour in contexts where we cannot enforce the agents to follow specific strategies.

Incentive mechanisms have gained great interest in the last decade and have been proposed for a wide area of applications. These applications range from information/data sharing to message relaying. Two domains in which the use of incentive mechanisms has become very popular are peer to peer networks [**?**, 43, 39, 41, 93] and electronic marketplaces [33, 59, 77, 92]. These are examples of communities where the participating agents are expected to participate well in order to be well accepted within the community.

Regarding the incentive mechanisms for peer to peer networks, in 2001, **Golle et al.** [20] provided a mechanism based on a micro-payment method. The authors considered that each peer is represented by an agent and that all agents which participate are economically rational. The aim of each agent was to maximize their expected utility based on: i) its beliefs about the actions that other agents will take, and ii) on its knowledge regarding the way that their payoffs are calculated. The main idea is based on the principle of *charge the agents for every download and reward them for every upload.* In particular, each agent i has a utility function U_i:

$$U_i = [f_i^{AD}(AD) + f_i^{NV}(NV) + f_i^{AL}(AL)] - [f_i^{DS}(DS) + f_i^{BW}(BW)] - FT \qquad (2.6)$$

where AD represents the amount downloaded, NV the network variety, DS the disk space used, BW the bandwidth used, AL the altruism (in case the agent derives utility from the satisfaction of contributing to the network), and FT represents the financial transfer (i.e., the amount of money the agent has to pay for using the network or the amount of money it might get paid for contributing to the network).

As we can see from equation 2.6 the utility function of an agent can increase by: i) the number of files it downloads, ii) the number of different options it has for downloading a particular file, and iii) the satisfaction that agents sometimes derive by contributing to the network. On the other hand, the utility function of an agent can decrease due to the cost of allocating disk space for files to be shared and to the cost of sharing the uploading bandwidth. The financial transfer FT factor is something that the agents might end up paying for using the network, or conversely for which they may end up getting paid in case they have contributed files to the network. Furthermore, the authors state an interesting observation that rare files should be treated differently from files that are more frequent in order to avoid discouraging agents to introduce new files. This is because if a node i decides to share a copy of a popular file, the likelihood of finding peers which are interested in downloading it is greater than if the node i adds a new file. In addition, the authors provide a very interesting analysis of their experiments, which shows that Napster suffers from the problem of free riders. As in [42], the proposed mechanism treats all the agents that enter the system equally. In our approach we are interested in considering systems and communities where the quality of each contribution is also taken into consideration.

In 2004, **Ma et al.** [42, 43] proposed the use of an incentive-based mechanism to determine the proper transfer bandwidth allocation for a peer that requests a file, based on its connection type, its utility function and its contribution. Unlike the mechanism proposed by Golle et al. [20], where the utility directly decreases each time the agent consumes the resources of the P2P network, in [42, 43] the utility is only influenced indirectly by the decrement of the download bandwidth the next time the agent requests a file. In particular, the utility function is:

$$U_i(\theta_i, x_i) = U_i(d_i, x_i) = \begin{cases} log(\frac{x_i}{d_i} + 1) & \text{if } x_i \leq d_i \\ log(2) & \text{if } x_i > d_i \end{cases} \tag{2.7}$$

where x_i represents the bandwidth allocated to node i, and is the maximal download bandwidth of node i. In order to calculate the bandwidth that will be allocated to node i the next time it will request a file, the following formula is used:

$$x(t) = argmax(\sum_{j \in R_i} U_j(\theta_j, x_j(t))) \tag{2.8}$$

where

$$\sum_{j \in R_i} x_j(t) \leq u_i. \tag{2.9}$$

where u_i is the utility of node i.

The authors' goal is to maximize the social welfare of a P2P network by making efficient use of P2P network resources, and to provide both fairness (e.g. distribute the load of the request from a few nodes to every node that participates) and incentives to contribute to all nodes in the community. In cases where the aggregated value of the download bandwidth is less than the aggregated value of the upload bandwidth, each agent will be simply assigned download bandwidth equal to its upload bandwidth. Similar to the approach presented by Golle et al. [20], this approach also does not consider the quality of contributions.

In 2003, **Yu et al.** [85] focused on the problem of relaying messages or providing information in a multiagent peer to peer network. In particular, when an agent, which represents a node, is interested in acquiring information from another node that is also represented by an agent, it sends a request to the agents of the neighbor nodes. For example, when an agent B receives a request for a service sent by a node A it can either answer it or if it is unable to provide the requested service, it can forward the request to other nodes. The problem that arises at this point is twofold:

- Why should a node provide a service to an unknown node?
- Why should a node provide a referral to an unknown node?

The first question refers to the case where agent B does have for example the file that agent A requested and the second question refers to the case where agent B cannot provide the requested service but has information regarding other nodes that might be more helpful. In order to address the above problem, the authors introduced a micro-payment method that rewards agents who either provide a service to another node or provide referrals to agents that might have the latter service. In particular, if the agent B had the service that agent A requested, then in order to provide it to agent A it must get a reward α. This reward will be deducted by the total currency that agent A has. Thus:

$$T_A = T_A - \beta$$

$$T_B = T_B + \beta$$

where T_A is A's balance, T_B is B's balance and β is the cost or reward for answering the query.

In case the agent B could not provide the requested service but it provided a referral to an agent C that might have it, then A would have to pay to B an amount equal to α and an amount equal to β to C as well. Thus:

$$T_A = T_A - \beta - \alpha$$

$$T_B = T_B + \alpha$$
$$T_C = T_C + \beta$$

where T_C is C's balance, and α is the cost or reward for providing one or more referrals. At this point we need to mention that each agent is assigned an initial budget equal to T. In order for an agent to be able to pay to search and acquire a particular service, it needs to have enough of a balance to cover the cost. Consequently, each agent is motivated to contribute either by offering services or by providing referrals, since this will increase its balance. Obviously, this approach treats all the services and all the referrals that are offered equally. This might not be always desirable since the quality of services might vary significantly while agents might provide inaccurate referrals. In order to alleviate this problem, the authors extended the above model by providing a more complicated dynamic pricing approach.

In 2005, **Cheng and Vassileva** [11] presented a peer to peer incentive based system for sharing articles in online e-learning communities. The authors addressed the problem that emerges when the users persist in overloading the system with low quality contributions, by presenting a system called Comtella. Their goal was to motivate the users to pay attention to the quality of information they choose to share in Comtella. The proposed mechanism highly rewards the acts of sharing a new topic and smoothly declines the reward as the number of contributed links in the community approaches a certain desired number. In addition, users can sponsor some of their links. In order to measure the quality of each contribution, the users are requested to evaluate the contributions of other users. The quality of a contribution is calculated as the summation of the ratings it received. Each time a user provides a rating for an article, he/she is awarded a certain number of *c-points* based on his/her reputation of giving high-quality postings. The *c-points* are essentially a type of virtual currency that the authors introduced. The user can use this currency to sponsor their links in order to increase their visibility in the search engine. The search engine has incorporated a mechanism that ranks the relevant (to the request) answers based on the numbers of *c-points* allocated by the contributors. The higher the position in the search list, the more likely the article is to be read. This is important, since the membership allocation of each user is associated with the ratings the user has acquired for providing articles. The different levels of memberships are associated with different rewards and privileges. The interesting point with this work is that the model not only considers the contribution of each agent, but also considers the quality of the contributions.

The idea of incorporating reputation into an incentive based community has been examined by other researchers. In fact, several approaches have been proposed in a wide area of applications. For instance, **Buchegger and Le Boudec** [8] proposed a reputation based approach that indirectly motivates the nodes in a mobile ad-hoc network to behave properly (i.e., to route and forward packages of other nodes), as well as assisting them to detect nodes that appear to be very prone to failures. More specifically, the authors consider that nodes can be classified as *misbehaving* and as *normal.* The general idea is to isolate nodes that are classified as *misbehaving* by denying to provide them with routing information or to forward their packages.

In particular, the system works as follows: each node i keeps a record with the behaviour of each node it has cooperated with. In order to calculate to which group to classify a node, it considers the beta distribution function with parameters (α,β):

$$\alpha := u\alpha + s$$

$$\beta := u\beta + (1 - s)$$

where $s = 1$ when the node misbehaved (e.g. dropped a package, significantly delayed a package etc) and 0 otherwise, while u is a discount factor that defines to what extent past experience should influence the future decisions. A node is classified as *misbehaving* when the beta function returns a value greater than a threshold, and is classified as *normal* otherwise. In addition, in cases where a node did not cooperate with a node in the past, it can utilize second hand information [8]. The proposed system uses reputation to detect potential malicious nodes and then it applies monitoring techniques for checking to what extent the reputation reflects the current reality regarding the participant node's behaviour.

In 2007, a similar problem was addressed by **Li et al.** [41]. More specifically, Li et al. addressed the problem of denying message relay in unstructured peer to peer networks. The proposed approach exploits a micropayment method in order to provide incentives to peers to forward or answer to messages sent by other peers. The querying peer (the *requestor*) sends a query message to some of its neighbors along with a promised reward. When a peer receives a message, it checks whether it can answer the message or not. If not, it may relay the message to its neighbors with a promised reward and so on. This procedure continues until a maximum number of hops or a node (known as "provider") is reached that can provide the requested information. If a provider is reached the information is reported backwards along the relay path and each node receives the reward it was promised. Clearly, a node will only receive a reward if it lies on a path between the requestor and a provider. Each node has incentives to relay a message, since this is the only way it can receive a reward, in cases where it cannot answer the query that the message contains. The authors validated their results through experimental evaluation and proved that their proposed approach brings to the system a utility higher than tradition approaches like breadth-first search or random walks. Furthermore, the authors provided a deeper analysis of their method and presented an approximation of the utility for the case of symmetric networks.

2.3 Scoring Rules & Graph-based Approaches in Information Evaluation

The areas of Statistics and Graph Theory have provided us with valuable tools to develop our solutions. In this section we first provide a detailed overview of scoring rules, a statistical tool that has been proposed to score predictions, a number of interesting approaches that are built on this concept, and then we present some work done in the area of information exchange that enables interactions with the use of graphs.

2.3.1 Scoring Rules

Scoring rules were proposed in 1971 by **Savage** [67] and have become very popular in information elicitation. In particular, the use of *scoring rules* focuses on evaluating a prediction in order to determine the rewards the predicator should receive. One of the most popular applications of scoring rules is the domain of forecast prediction. More specifically, given a prediction q that an event A will happen, a score $S(q,p)$ is computed. The parameter p is 1 if the event A happens and 0 otherwise. In general, in case there is a set O of possible events and Q is the probability distribution over O, the scoring rule is defined as $S : O \times Q \mapsto R$. For instance, consider that a forecaster is asked to provide a prediction whether it will rain tomorrow and she predicts that with probability q it will rain. In this case $O = \{\textit{rain, not rain}\}$, $Q(\textit{rain}) = q$ and $Q(\textit{not rain}) = 1 - q$. The expected reward the predictor will receive is defined as:

$$u(q,p) = p \cdot S(q,1) - (1-p) \cdot S(q,0)$$

where $S(q,1)$ is the score if it rains, $S(q,0)$ is the score received if it does not rain and p is 1 if it rains and 0 otherwise. Three of the most well known scoring rules are:

1. Quadratic (or Brier)
$$S(q,p) = 1 - (p-q)^2$$
2. Logarithmic
$$S(q,p) = log(q)$$
3. Spherical
$$S(q,p) = \frac{p}{\sqrt{q^2}}$$

where p is equal to 1 if the event happens (i.e., if it rains) and to 0 otherwise (i.e., if it does not rain).

In our work, we are interested in *strictly proper scoring rules.* Strictly properness is a key property. The strictly proper scoring rules are scoring rules that are uniquely maximized when the reported prediction is equal to the real prediction (i.e., only when $p = q$, where p is the observed probability and q the reported). This property is important since it can provide appropriate incentives to enforce a truthful behaviour. Note, that the quadratic, the logarithmic, and the spherical scoring rules are all *strictly proper.*

Zohar and Rosenschein [92] propose the use of scoring rules for designing a payment mechanism for the context of information elicitation. The expected utility of an agent providing an information that has a real value x, which takes values in the domain W, is:

$$U(x, w) = \sum_{w \in W} p_{w,x} \cdot u_{w,x} - c$$

where c is the cost for acquiring the information, x is the real value of the variable, w is the value the agent reports and $p_{w,x}$ is the probability that the agent will report a value equal to w when the real value is x. The three constraints the payment mechanism should satisfy are:

1. **Truth telling.** Every agent should have the incentive to reveal its true value x (i.e., the reward that it will receive by providing the true information should be higher than the reward it will receive by providing an untruthful information). Formally: $\forall x, x'$ such that for $x \neq x'$

$$\sum_{w} p_{w,x} \cdot (u_{w,x} - u_{w,x'}) > 0$$

2. **Individual Rationality.** An agent should have a positive expected utility (i.e., the reward that it will receive by providing information should be higher than the cost for acquiring the information).

$$\sum_{w} p_{w,x} \cdot u_{w,x} > c$$

3. **Investment.** The value of information should be greater than the cost (i.e., any guess x' the agents makes (without actually computing the real value of x) should be less profitable than paying to discover the real value of the variable).

$$\sum_{w \in W} p_{w,x} \cdot u_{w,x} - c > \sum_{w \in W} p_{w,x} \cdot u_{w,x'}$$

Zohar and Rosenschein [92] take advantage of scoring rules in defining the value function $u_{w,e}$. In particular, the proposed $u_{w,e}$ is a linear transformation of a scoring rule:[4]

$$u_{w,e} = \alpha \cdot S(Pr(w|x), w) + \beta_w$$

where S can be any of the following scoring rules:[5]

1. Quadratic (or Brier)
$$S(p, w) = 2p_w - \sum_{w'} p_{w'}^2$$

2. Logarithmic
$$S(p, w) = log(p_w)$$

3. Spherical
$$S(p, w) = \frac{p_w}{\sqrt{\sum_{w'} p_{w'}^2}}$$

The probability p_x is the reported probability that the event will take the value x. The authors show that by carefully selecting the values of α and β their mechanism satisfies both individual rationality and investment constraints.

At this point we would like to stress that the above three families of scoring rules are not equivalent. For instance, the spherical and the quadratic scoring rules are both symmetric. A scoring rule is considered to be symmetric if the score remains the same when the probabilities that are assigned to the correct answers are permuted without changing their values. Thus, settings where symmetry is important restrict us from applying the logarithmic scoring rule.

In particular, they show that in order for the investment constraint to be satisfied it is sufficient to select an α such that the following inequality holds for every x':

$$\alpha > max[\frac{c}{\sum_{w,x} p_{w,x}(S(Pr(w|x), w)) - S(Pr(w|x'), w))}]$$

while for the individual rationality the following constraint is sufficient:

$$\beta_w = \beta > c - \alpha \sum_{w,x} p_{w,x}(S(Pr(w|x), w))$$

[4] The affine transformation of a strictly proper scoring rule is also strictly proper.

[5] Note that this is a more generalized definition of the Quadratic, Logarithmic and Spherical Scoring Rules, as the different values the variable w can have an event can be more than two.

Furthermore, they introduce the concept of robust mechanisms to address the problem of belief variation among different sellers, and they provide algorithms which can learn the robustness level of a given payment scheme.

In 2005, **Miller and Resnick** [46] addressed the problem of feedback elicitation when an independent and objective outcome is not available. The authors presented a scoring rule based mechanism inspired by methods that define the score of a peer by comparing the similarity of the rating it provided with the ratings that other peers provided. The authors stressed that the previous methods are prone to malicious behaviour. In particular, let s be a seller that has strategically built a good reputation. If a buyer b that was deceived by s knows that the reputation of the seller is high it might be reluctant to report his real experience with s given that this will differ from the experiences of other buyers and thus might choose to provide a rating similar to what other buyers provide. This will cause the reputation of the seller to remain high giving the ability to s to deceive more buyers in the future.

In order to address this problem, Miller and Resnick suggested a method in which a rater is not scored based on the deviation of the ratings but based on the likelihood assigned to the rater's possible rating and the rater's actual rating. More specifically, each rater i receives a signal s_i based on the type t of the product for which it announces a rating r_i. The score a rater j will receive can be defined using one of the following strictly proper scoring rules:

1. Quadratic (or Brier)
$$R(q,p) = 2g(s^i_h|r^i) - \sum_{h=1}^{M} g(s^i_h|r^i)^2$$
2. Logarithmic
$$R(q,p) = log(g(s^i_h|r^i))$$
3. Spherical
$$R(q,p) = \frac{g(s^i_h|r^i)}{\sqrt{g(s^i_h|r^i)^2}}$$

where $g(s^j_h|r^i)$ is the probability that the rater j will receive signal h given that rater i announced r^i. Thus, essentially they transfer the problem of honest reporting in determining the possible ratings the raters provide.

In 2007, **Jurca and Faltings** [33] proposed a collusion-resistant incentive compatible feedback payment mechanism. The authors considered an electronic-market setting where rational agents who act as buyers experience the same product. The quality of a product defines its type θ and remains fixed. Each buyer submits a feedback. Each time a feedback is submitted the buyer receives a payoff which is determined by the reports other agents submitted and the probability distribution over the real quality of the product. The expected payoff for an agent i is:

$$V(a_i, a_{-i}|o_i) = \sum_{n=0}^{N-1} Pr[n|o_i] \sum_{x=0}^{N-1} \pi[n|a_{-i}]\tau(a_i(o+i), x)$$

where o_i is the true observation, a_{-i} is the strategy profile of the rest of the agents, $\tau(a_i(o+i), x)$ is the amount paid by i given that the number of positive reports is equal to x and can be determined based on scoring rules [46], $\pi[n|a_{-i}]$ is a_i's belief about the distribution of the reference reports, and $Pr[n|o_i]$ is the probability that exactly n positive reports were submitted by the other $N-1$ agents and can computed as follows:

$$Pr[n|o_i] = \sum_{\theta \in \Theta} Pr[n|\theta] Pr[\theta|o_i]$$

where Θ is the set of all possible types, $Pr[n|\theta]$ is given by the binomial distribution, and $Pr[\theta|o_i]$ can be computed from Bayes' Law. Essentially, the authors consider that each buyer will reason about whether she liked the product she purchased based on a probability distribution on the possible types of the product.

In 2008, **Papakonstantinou et al.** [50] proposed a method for addressing the problem of probabilistic estimates elicitation for forecast prediction in settings where the cost function is unknown. They presented a two phase mechanism that incorporates a payment approach based on scoring rules. During the first phase, each agent a_i is requested to submit its expected cost function $\hat{c}_i(\cdot)$ (for eliciting the probabilistic estimates) with a required minimum precision θ_0. The forecast is assigned to the agent a_m that provided the lowest cost for the given precision θ_0. Formally:

$$\hat{c_m}(\theta_0) = min_{k \in \{1,\ldots,N\}} \hat{c_k}(\theta_0)$$

During the second stage the scoring rule is announced. The agent a_m produces an estimate x with a precision θ and reports $\hat{x}$ and $\hat{\theta}$ and after the actual outcome is observed the payment of the agent a^* is computed. In particular:

$$P(x_0; x, \theta) = \alpha \cdot S(x_0; x, \theta) + \beta$$

where x_0 is the actual outcome observed and the parameters α and β are computed as follows

$$\alpha = \frac{c'(\theta_0)}{\bar{S}'(\theta_0)} \text{ and } \beta = c(\theta_0) - \frac{c'(\theta_0)}{\bar{S}'(\theta_0)} \bar{S}(\theta)$$

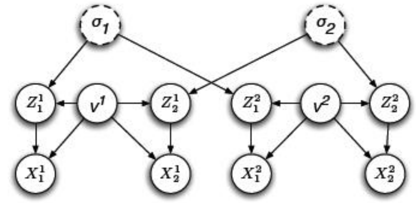


Figure 2.1: Example of the Bayesian Network used in [22]

but based on the second-lowest reported cost function. The authors proved that both the stages are incentive compatible, while the mechanism is individual rational and they argued that their proposed mechanism provides strong incentives to make an estimate with a precision at least equal to θ_0.

2.3.2 Graph-based Approaches in Information Evaluation

Graph theory has been widely used in describing social networks as well as interactions of agents. A number of approaches for determining the popularity of a node in a graph have been proposed.

In 2009, **Hendrix et al.** [22] proposed a Bayesian-based mechanism that exploits previous advice in order to evaluate the truthfulness of information providers. An example of a Bayesian network that describes the information provided by two agents a_1 and a_2 over a painting is depicted in Figure 2.1. For clarity reasons we will refer to this interaction as the interaction a. The nodes v^1 and v^2 represent two paintings assigned for appraisal. The nodes X_1^a and X_2^a represent the two estimates from the information providers. The nodes Z_1^1 and Z_2^1 represent the estimated deviation of the ratings based on past interactions while X_1^b, X_2^b, Z_1^b and Z_2^b represents equivalent nodes for another interaction b. The nodes σ_1 and σ_2 represent the capabilities of the agents and their values are unknown. Given that the network presented in Fig. (2.1) is a Bayesian network, the edge between two nodes indicates that the source node is the direct cause of the target node. For example, the estimate X_1^a depends on both the error in agent's a_2 estimation Z_2^a and the true value v^2 of the painting. The size of the network expands linearly to the number of provider types, while there will be nodes $((X_1^i, ..., X_1^i), V^i)$ for each interaction i. The Bayesian network is used to calculate the capabilities σ_i of each agent a_i. Furthermore, the authors suggest a Bayesian model for determining the strategy of untruthful information providers as well as a Bayesian network for determining the appraisal of a good based on the provided reports. The main use of the latter Bayesian network is to determine the weight of the report that each of the information providers provided. The authors performed a set of experiments and demonstrate that their model outperforms all the other models in the AAMAS 2008 *A*gent *R*eputation and *T*rust (known as *ART*) testbed competition. The ART testbed is a competition where the agents need to appraise the value of paintings from different areas and might range from $1,000$ to $100,000$.

In 2009, **Peng et al.** [51] proposed a graph based method for the analysis of the interactions among agents in multiagent systems. More specifically, the authors are interested in presenting a method that can provide a clear picture of the interactions among the agents through the recognition of possible communication patterns of events. Note that simply depicting the interactions of each agent results in a very complicated graph the processing of which may require computationally expensive algorithms. The proposed approach demonstrates that carefully defining a graph for representing interactions can overcome a significant level of complexity issues without critically compromising the quality of the results. The authors consider a graph in which the nodes represent events that are triggered by various agents. The edges represent messages that are sent during the execution of an event. Each node has a label that depicts the name and the activity that executed the corresponding event. The information of each event is captured on the label of the edge that represents it. Each event is triggered by exactly one message. The main contribution is that they show that by defining in this way the graph that depicts the interactions between events and messages, the graph that occurs is essentially a forest of trees. From the time complexity perspective this is very important since there are a number of algorithms that can be efficiently applied to trees to capture common patterns.

In our case we use a graph to depict the consistency of information providers that disseminate information regarding the behaviour of their agents in a way that both identifies truthful providers but also promotes honesty for future interactions.

Chapter 3

The Architecture

In this chapter we provide the notation that we will be using, the overall architecture of our proposed approach and a more formal definition of the three problems we address in this thesis. Before we proceed, we restate the overall problem towards which the solution of the latter three problems contributes.

The overall problem we aim to address is as follows: *How can we discourage the existence of malicious agents and agents who make poor contributions in peer-based communities but also encourage the participants of these communities to further improve their contributions?*[1]

3.1 The Model

Let C_i denote community i and let a_j denote agent j. Each community C_i:

- offers a set of services S_i (e.g. purchasing items, downloading media files etc),
- has a budget B_i that it can use to purchase information about agents from other communities, and
- maintains a history H_i of its past interactions with other communities (Fig 3.1).

We assume that if a_j is a member of community C_i then C_i can observe and judge the quality of agent a_j. In particular, we assume that community C_i maintains an *evaluation model* for all member agents, and is able to assign an *evaluation rating* r_j^i to agent a_j,

[1] Recall, by peer-based communities, we refer to communities in which a significant portion of the services they offer are based on the contributions of their users.

where r_j^i is some real number in the interval $[\alpha, \beta]$, $0 \leq \alpha < \beta$. The evaluation rating can be either a reputation rating or a trust rating. Recall, by *trust* we refer to the degree an agent honored its promise and by *reputation* we refer to the degree of satisfaction (i.e., contribution to the maximization of their utility) the agent brought to the other agents with which it has interacted.

Since communities may use different evaluation models, or may interpret ratings differently, they are also requested to provide the *type*, θ_j^i of the agent a_j when exchanging information about an agent with another community. We assume that $\theta_j^i \in \{\text{good}, \text{poor}\}$, and that this is the *interpretation* that community C_i makes concerning its rating r_j^i for agent a_j. In particular, an agent can be either a good contributor or poor contributor. Finally, by $M(a_j)$ we denote the set of communities that the agent a_j has participated in. We will refer to these communities as the *advisor communities* and to C_i as the *recipient community.* When the recipient community C_i receives a request from a prospective agent a_j, it communicates with the set $M(a_j)$ of advisors communities and requests information about a_j's past behaviour in exchange for a payment.

In particular, in our setting, each agent a_j can:

- participate in more than one community at a time,
- join a new community,
- withdraw from a community in which it already participates.

Each community C_i can:

- allow an agent to join with certain privileges,
- deny entrance to an agent,
- provide information about its agents to other communities and receive a payment from each of these communities,
- request information about prospective agents from other communities and provide them with a payment.

3.2 The Architecture

A key element of our mechanism is that a community C_i that is interested in acquiring information about an agent a_j does not necessarily have to ask every community in $M(a_j)$. In fact, it should select a subset $m(a_j)$, $m(a_j) \subseteq M(a_j)$, and ask the communities in $m(a_j)$. The information each community $C_k \in m(a_j)$ has to provide consists of two parts:

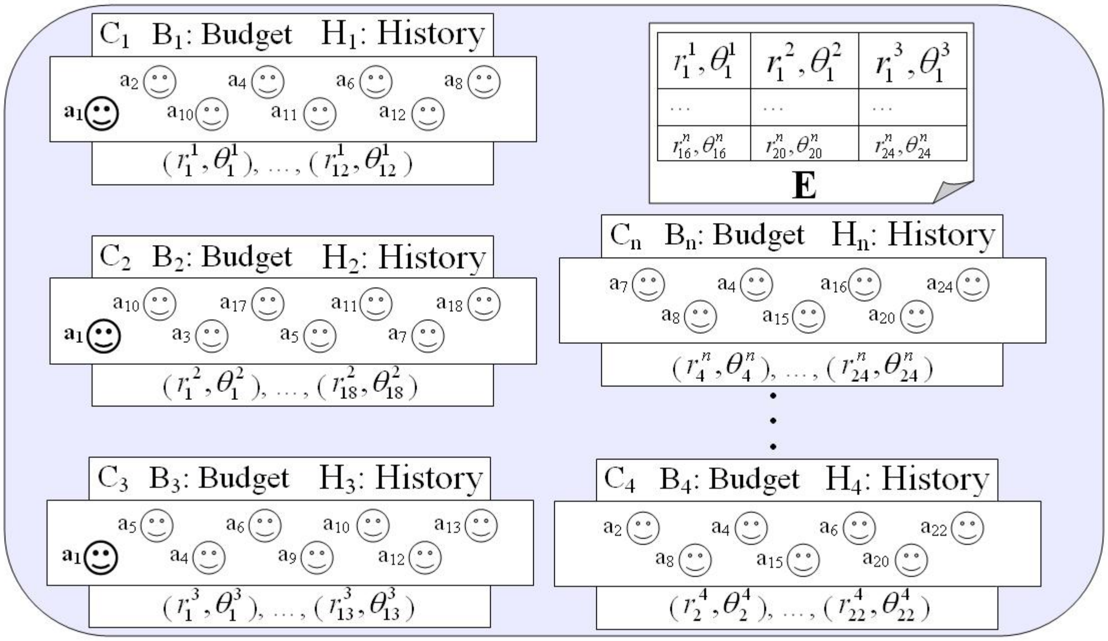

C1 B1: Budget H1: History
a2 a4 a6 a8
a1 a10 a11 a12
$(r_1^1, \theta_1^1), \ldots, (r_{12}^1, \theta_{12}^1)$
C2 B2: Budget H2: History
a10 a17 a11 a18
a1 a3 a5 a7
$(r_1^2, \theta_1^2), \ldots, (r_{18}^2, \theta_{18}^2)$
C3 B3: Budget H3: History
a5 a6 a10 a13
a1 a4 a9 a12
$(r_1^3, \theta_1^3), \ldots, (r_{13}^3, \theta_{13}^3)$
r_1^1, θ_1^1 r_1^2, θ_1^2 r_1^3, θ_1^3
... ...
r_{16}^n, θ_{16}^n r_{20}^n, θ_{20}^n r_{24}^n, θ_{24}^n
E
Cn Bn: Budget Hn: History
a7 a4 a16 a24
a8 a15 a20
$(r_4^n, \theta_4^n), \ldots, (r_{24}^n, \theta_{24}^n)$
C4 B4: Budget H4: History
a2 a4 a6 a22
a8 a15 a20
$(r_2^4, \theta_2^4), \ldots, (r_{22}^4, \theta_{22}^4)$

Figure 3.1: Example of communities

the evaluation rating r_j^k and the type θ_j^k of the agent a_j. As we will see later, essentially the type θ_j^k represents the interpretation of the rating r_j^k, since different communities might use different evaluation models. For example a rating 0.6 in one community might represent an agent which is a good contributor while in another community it might represent an agent which is a poor contributor. In exchange for information, community C_k receives a payment. This payment depends on how *valuable* the recipient community finds the information C_k provided.

The importance of the information that each community provides is measured by a function $I(x, \hat{r}, y)$, $I(x, \hat{r}, y) : [0,1] \times [0,1] \times \{good, poor\} \mapsto \mathbb{R}$, which we will refer to as the *i-Function* and to its value as the *i-Score*. The *i-Function* $I(x, \hat{r}, y)$ considers both:

- the degree to which an evaluation rating x deviates from the evaluation rating $\hat{r}$ which is determined by the recipient community after experiencing the agent, and
- the direction towards which the rating x influences the recipient's decision (i.e., to accept or reject an agent).

For each tuple (r_j^k, θ_j^k) that a community C_k provides it receives an *i-Score* $I(r_j^k, \hat{r}, \theta_j^k)$, where r_j^k and θ_j^k are the evaluation rating and the type of the agent in the community C_k, respectively. Based on the latter *i-Score*, we determine the payment of the community C_k.

In particular, we consider a payment function $P(I(r_j^k, \hat{r}, \theta_j^k))$, $P : \mathbb{R}^+ \mapsto \mathbb{R}$, where $\hat{r}$ is the evaluation rating of the agent by the recipient community. As we will explain in detail in Chapter 5, only the communities in $m(a_j)$ which characterized the agent with a type equal to $\hat{\theta}$ (i.e., the type that the recipient community assigned to the agent) will get paid.[2]

Only after this decision is made do the communities in $m(a_j)$ get paid for the information that they provided. Since the payment each advisor community will receive is based on the rating the recipient community provides, the recipient community can try to manipulate it by reporting an untruthful rating. In order to prevent the latter from happening we consider the existence of a trusted entity E which maintains a *Selling Board* and through which the payment exchange takes place. Each community $C_i \in C$ maintains on E's selling board a tuple (r_j^i, θ_j^i) for each of its agents a_j, where r_j^i and θ_j^i are the evaluation rating and type of agent a_j in C_i, respectively. Note that the information regarding the type of an agent is vital. This is due to the fact that communities may be using different evaluation models and thus a rating r' in one community may describe a *poor contributor* while in another a *good contributor*. In general, a community will decide to accept a good agent and reject a bad one (i.e., a *poor contributor*).

If a community C_i tries to manipulate the distribution of the payments (i.e., provide higher payments to particular communities and lower to other communities) by posting on E's selling board a rating different from the one it experiences, then each time another community C_j buys this information it will receive an inaccurate rating, and thus both the payment and the probability that C_j will ask C_i again in the future will decrease. Given that the number of the communities and the identity of the communities that will experience the latter inaccurate information is unknown, C_i may put itself at a significant disadvantage.

In more detail, the overall architecture is as follows: let C_i be a community that wishes to acquire information regarding an agent a_j from a set of communities $m(a_j)$ where $m(a_j) \subseteq M(a_j)$. The community C_i sends a request to the entity E and authorizes it to deduct an amount equal to w from its budget B_i, where $w < B_i$. Upon receiving the request, E replies with the requested evaluations of the agent. After E sends the information, the arrangement of the amount that each community in $m(a_j)$ should receive takes place.

The amount w will be distributed based on the *i-Score* that each advisor community $C_k \in m(a_j)$ receives based on the information it provided. If the recipient community C_i accepts the agent a_j, it will wait for a time period T and then will post its evaluation rating

[2] If community C_k decides to *reject* agent a_j, then the payment procedure is a little more complicated since the community never gets the opportunity to directly observe and evaluate agent a_j. In this case the type of the agent is set to *poor* and its rating $\hat{r}$ is determined based on the reported values of the information providers. A detailed description on how exactly the payment takes place is presented in Chapter 5.

$\hat{r}$ and type $\hat{\theta}$ about agent a_j on E's selling board. Based on the reported value, the *i-Score*, $I(r_j^k, \hat{r})$, each advisor community receives will be calculated. Based on the *i-Scores* the entity E will distribute the above amount to the communities that classified the agent a_j with the same type as the one used by the recipient community to classify the agent (i.e., *good* or *poor*). Note that each community can update the evaluation of its agents after the expiration of a time period T set by the trusted entity E. If community C_i decides to *reject* agent a_j, then the payment procedure is a little more complicated since the community never gets the opportunity to directly observe and evaluate agent a_j. Instead, community C_i sets the type of agent a_j to be $\theta_j^i = \text{poor}$, and computes $\hat{r}$ to be the average evaluation rating of all communities in $m(a_j)$ that also assigned type $\theta_j^k = \text{poor}$.

Each community can choose either to post the evaluation ratings of its agents based on its belief about what the recipient communities will experience or to simply post the evaluation rating it experiences with the hope that potential recipients will have a similar experience. We argue that each of the above choices is acceptable since a recipient community is interested only in acquiring information that can help it make a correct decision regardless of the motivation of the provider.

Summarizing, a community C_i that is interested in acquiring information about a_j:[3]

1. C_i acquires the set $M(a_j)$ (Fig. 3.2)
2. C_i determines the set $m(a_j)$ based on past experience with the communities in $M(a_j)$ (Fig. 3.3).
3. C_i acquires the tuple (r_j^k, θ_j^k) from each community C_k in $m(a_j)$ (Fig. 3.4).
4. C_i decides whether to accept the agent a_j or not
    - if a_j is accepted, C_i evaluates a_j and posts the tuple (r_j^i, θ_j^i) on E's selling board (Fig. 3.5).
5. C_i determines the payment each community C_k will receive (Fig. 3.6).

3.3 The Problems

In the above procedure we assume that the set $m(a_j)$, the *i-Function* $I(x, \hat{r})$, and the evaluation ratings r_j^k are all known. The way that these three parameters are determined constitutes the three problems we address in this thesis. In this section we provide a formal definition of each of these three problems and we state our main assumptions.

[3] The black boxes in figures 3.3, 3.5 and 3.6 indicate the problems we address.

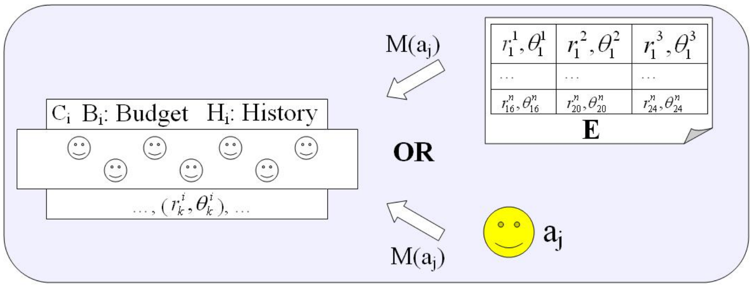

M(a$_j$)
r_1^1, θ_1^1
r_1^2, θ_1^2
r_1^3, θ_1^3
…
r_{16}^n, θ_{16}^n
r_{20}^n, θ_{20}^n
r_{24}^n, θ_{24}^n
E
C$_i$ B$_i$: Budget H$_i$: History
OR
…, (r_k^i, θ_k^i), …
M(a$_j$)
a$_j$

Figure 3.2: Selecting advisor set $M(a_j)$

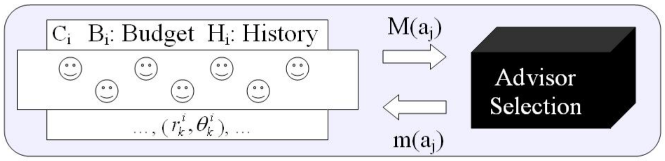

C$_i$ B$_i$: Budget H$_i$: History
M(a$_j$)
Advisor Selection
…, (r_k^i, θ_k^i), …
m(a$_j$)

Figure 3.3: Selecting advisor subset $m(a_j)$

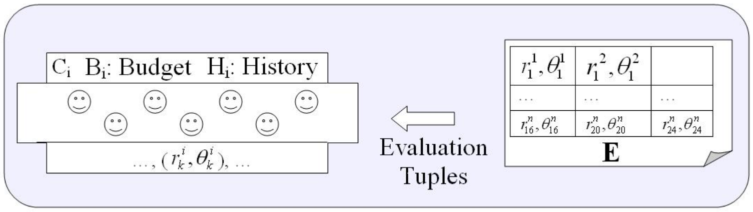

C$_i$ B$_i$: Budget H$_i$: History
r_1^1, θ_1^1
r_1^2, θ_1^2
…
r_{16}^n, θ_{16}^n
r_{20}^n, θ_{20}^n
r_{24}^n, θ_{24}^n
E
Evaluation Tuples
…, (r_k^i, θ_k^i), …

Figure 3.4: Acquiring evaluation tuples

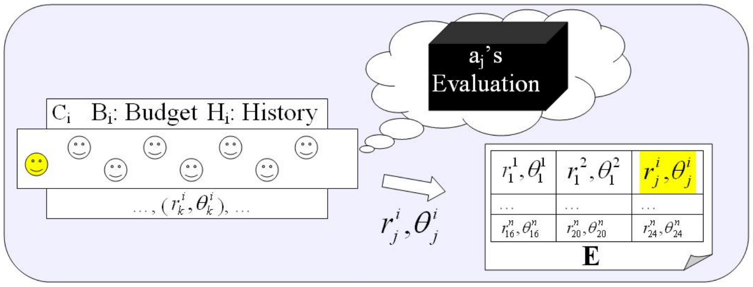

a$_j$'s Evaluation
C$_i$ B$_i$: Budget H$_i$: History
r_1^1, θ_1^1
r_1^2, θ_1^2
r_j^i, θ_j^i
…
r_{16}^n, θ_{16}^n
r_{20}^n, θ_{20}^n
r_{24}^n, θ_{24}^n
E
…, (r_k^i, θ_k^i), …
r_j^i, θ_j^i

Figure 3.5: Evaluating agent a_j

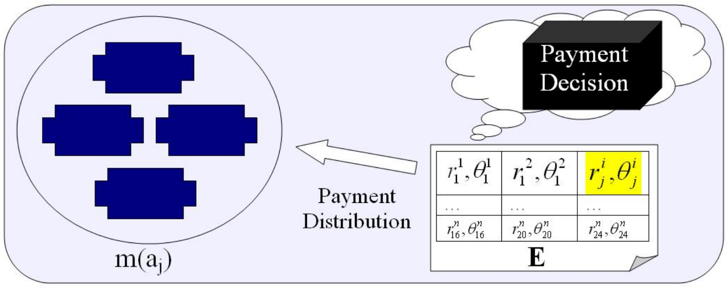


Figure 3.6: Payment Decision

Problem 1. Advisor Selection: *Consider a set of self-interested communities C that report private information and a number k. Design a mechanism to determine a subset V, where $V \subseteq C$ and $|V| = k$, such that the communities in V have strong incentives to provide truthful reports.*[4]

A novel approach for determining the set V that satisfies the above two requirements is presented in Chapter 4. The key idea of the proposed solution lies in the exploitation of the consistency of the advice the communities in V have provided to C_i in the past. This is because an advisor community can learn the probability other communities can lie and thus it can reason in which cases it can afford to lie. However, if the selection also takes into consideration the extent an advisor community is consistent with other truthful communities then the uncertainty about the reports these community provide creates strong incentives for honest reporting.

At this point we need to remind the reader that we consider that an advisor community is honest when the evaluation rating it provided appears to be similar with the rating that the recipient community experiences. Briefly, by *honest* we refer to a community that provides correct information while by *dishonest* we refer to a community that provides incorrect information.

Problem 2. Payment Decision: *Let (r_i, θ_i) be the evaluation tuple provided by a community C_i, where $r_i \in [\alpha, \beta]$ and $\theta_i \in \{0, 1\}$. Given a set of tuples $D = \{(r_1, \theta_1)..., (r_n, \theta_n)\}$ and a rating $(\hat{r}, \hat{\theta})$, where $\hat{r} \in [\alpha, \beta]$, $\hat{\theta} \in \{0, 1\}$ and $\alpha \geq 0$, determine the payment $P(r_k, \hat{r})$ that each community C_i in S should receive, with respect to the accuracy and effectiveness of the tuple (r_i, θ_i) it provided.*

Towards the solution of this problem we first focus on evaluating the accuracy and the helpfulness of the tuples in D through a scoring function that we call *i-Function*. In partic-

[4] By *similar history* we refer to the case where the probability that each community will provide a truthful report lies in a range in which the highest value does not provide significantly more important information than the lowest value.

ular, in Chapter 5, we suggest a novel approach for how to derive an appropriate *i-Function* $I(r_i, \hat{r})$ that evaluates the provided tuple (r_i, θ_i) by both considering the rating r_i and the type θ_i (i.e., the interpretation of the rating). Briefly, let $\hat{r}$ be the rating a community experiences after accepting an agent a. If the rating that an advisor community reported deviates from $\hat{r}$ and influences the recipient community towards making the wrong decision (i.e., to decline a good contributor or to accept a poor contributor), the advisor community should receive a lower payment than if it had reported a rating of equal deviation but which was influencing towards the right decision (i.e., accept a good contributor or decline a poor contributor). For example, consider $\hat{r} = 0.7$ and the agent proved to be good. If two advisor communities had reported that the agent is good and its ratings are 0.6 and 0.8, respectively, then the second advisor should receive a higher payment than the first. The reason is that the first rating is more likely to be the result of a deceptive behaviour than the second one.

Problem 3. Trust Modeling: *Let C_i be a community that requests an agent a_j to commit contributing a set of services S_i. Determine a_j's trustworthiness based on the extent it honored its commitment.*

Our suggested solution for the *Trust Modeling Problem* is presented in Chapter 6. The main contribution of our approach is the presentation of the minimum requirements for a promised-based trust model for settings in which crucial collective decisions are made based on the contributions that each participant declares (i.e., planning, coalition formation, etc.). In particular, we target settings in which the exact knowledge of the contribution is important.

Last, we would like to state our main assumptions. In particular, we assume that

- each community is interested in acquiring information from at least a subset of the communities which participate in the mechanism,
- each community evaluates its agents by using both
    - a continuous rating, which it converts to a commonly agreed upon range $[\alpha, \beta]$, and
    - a binary rating that indicates whether the continuous rating corresponds to an agent which is either a good contributor or a poor contributor,
- each community trusts and recognizes the authority of the trusted entity E,
- each agent can be uniquely identified.

We argue that all of the above assumptions are realistic. In addition we would like to note that the practical implementation (for example, of the trusted entity E) is beyond the scope of this thesis [75].

Chapter 4

The Advisor Selection Problem

In this chapter we address the problem of exchanging evaluation ratings between communities. These ratings can either depict the trustworthiness or the reputation of the entity they refer to. More specifically, we consider a repeated setting and we propose the use of a graph-based heuristic which exploits the consistency among the advice of candidate advisors. Our approach reduces the number of communities that are queried each time a community seeks information about a prospective agent in such a way that all the advisor communities are inclined to provide truthful reports. This reduction is important since each time a community requests information from another community it has to provide some resources. Thus, by limiting the number of advisors the community contacts we limit the resources it will have to spend. We argue that exploiting consistency among good advisors will further promote honest behaviour since the exchange of the resources will get restricted among the most reliable sources. This will create a more competitive environment between the advisor communities which will be inclined to be truthful in order to maintain or increase the probability of being asked in the future.

4.1 The Model

Let C be the set of all communities and A the set of all the agents that participate in the communities of C. We assume that if an agent a_j is a member of community C_i then C_i can observe and judge the quality of agent a_j. In particular, we assume that each community C_i maintains an *evaluation model* which it uses to assign an *evaluation rating* r_j^i to agent a_j, where $r_j^i \in \{Good, Poor\}$. A rating equal to *Good* represents an agent which has good behaviour inside the community C_i (e.g. a good contributor in providing files, a good seller), while a rating equal to *Poor* represents an agent that has poor behaviour (e.g. malicious behaviour, contribution of low quality files etc.).

Each time an agent a_j wishes to join a community C_i, C_i contacts the communities in which a_j is currently, or was previously, a member, and requests information regarding a_j's behaviour. We denote the set of these communities as $S(a_j)$ and we refer to the community C_i as the *recipient community* and the communities in $S(a_j)$ as the *advisor communities.*

Currently, we assume that the set $S(a_j)$ is provided by the agent a_j. In exchange for information, each advisor community receives a payment from community C_i. This payment depends on how *useful* or *important* community C_i finds the information provided by the advisor. The expected utility $EU(C_i)$ of a community C_i with respect to the information exchange mechanism is defined as:

$$EU(C_i) = Value(i) - Cost(i) \tag{4.1}$$

where

$$Value(i) = Pmnt^i_{provide} \cdot Pr^i_{provide} \text{ and } Cost(i) = Pmnt^i_{request} \cdot Pr^i_{request} \tag{4.2}$$

By $Pr^i_{provide}$ we denote the probability the community C_i will *provide* information about its agents to other communities and by $Pmnt^i_{provide}$ we refer to the expected payment the community C_i will receive from these communities. By $Pr^i_{request}$ we denote the probability an agent a^* with previous history will *request* to enter the community C_i, and thus the community C_i will be interested to *request* information about this agent, while by $Pmnt^i_{request}$ we denote the expected payment the community C_i will have to provide to *request* information for a^*.

Each community C_i is interested in maximizing its utility $EU(C_i)$. This can be achieved by maximizing the probability $Pr^i_{provide}$ and the payment $Pmnt^i_{provide}$ and by minimizing the payment $Pmnt^i_{request}$. The community C_i does not have a direct control over the probability $Pr^i_{request}$ since it cannot control whether other agents will be interested to join it. More specifically, the community C_i does not have control over the upper bound of $Pr^i_{request}$. However, it is to C_i's best interest to maintain a higher upper bound for $Pr^i_{request}$ since this will give C_i more flexibility when accumulating information for prospective agents.

In this chapter, we provide a mechanism that maximizes both the probability $Pr^i_{provide}$ and the payment $Pmnt^i_{provide}$ C_i receives, while it also minimizes the payment $Pmnt^i_{request}$ the community C_i has to provide when it requests information about an agent. We minimize the payment $Pmnt^i_{request}$ by proposing an approach that considers the selection of a subset L of candidate advisors without compromising the quality of information C_i receives while at the same time our method allows the communities to maximize the probability $Pr^i_{provide}$ when they have an honest behaviour. The payment $Pmnt^i_{request}$ will be also maximized if the complimentary payment method that is used with our mechanism maximizes the rewards when an honest reporting is given.

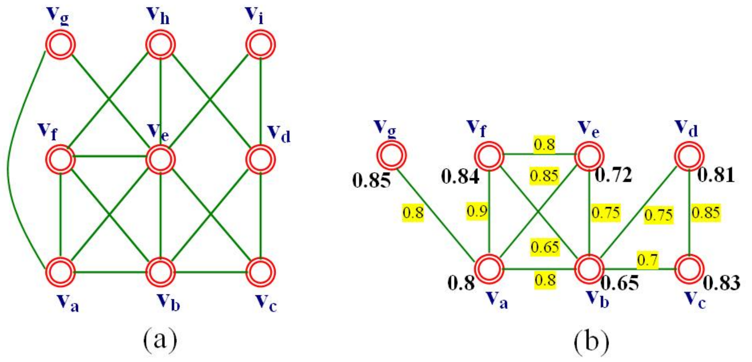


Figure 4.1: Example of a) Consistency Graph G_i and b) Consistency Subgraph with $Pr^{+}(V)$ and $Pr^{+}(E)$ over their vertices and edges

v	$P^{+}(v)$	$P^{-}(v)$	$W(v)$
v_a	0.8	0.5	15
v_b	0.65	0.75	17
v_c	0.83	0.8	20
v_d	0.81	0.6	31
v_e	0.72	0.6	24
v_f	0.84	0.5	30
v_g	0.85	0.8	18
v_h	0.65	0.7	3
v_i	0.5	0.5	15

Table 4.1: G_c's weights over vertices

The main idea for the selection of set L of advisor communities lies on the exploitation of the consistency among communities, which provide good quality information. Our heuristic is flexible enough to consider both the number of advisors a community is consistent with and also the quality of the consistency. Briefly, we consider two or more communities to be *consistent* when they provide similar information.

We acknowledge that the full utility of a participant community may depend on many features not modeled by the exchange utility $EU(C_i)$, such as the gain for not sharing information. Our focus in this work is on communities which have decided that the utility gained by exchanging information is higher than the utility they would have gained by not participating in the exchange mechanism.

e	$P^+(e)$	$P^-(e)$	$W(e)$
(v_a, v_b)	0.6	0.55	20
(v_a, v_e)	0.85	0.9	15
(v_a, v_f)	0.9	0.8	17
(v_a, v_g)	0.8	0.5	15
(v_b, v_c)	0.7	0.7	19
(v_b, v_d)	0.75	0.5	16
(v_b, v_e)	0.8	0.6	12
(v_c, v_d)	0.85	0.8	14
(v_c, v_e)	0.5	0.6	13
(v_d, v_h)	0.8	0.55	16
(v_d, v_i)	0.7	0.9	22
(v_e, v_f)	0.85	0.5	17
(v_e, v_g)	0.55	0.45	9
(v_e, v_h)	0.85	0.6	18
(v_e, v_i)	0.65	0.8	18
(v_f, v_h)	0.8	0.54	19

Table 4.2: G_c's weights over edges

4.2 The Advisor Selection Algorithm

4.2.1 Definitions

In order to exploit the consistency among the information the advisor communities offer we consider a graph-based heuristic. Each community C_i constructs a graph G_i which maintains partial information regarding the consistency among the information that other communities have provided to C_i in the past. In particular, we refer to the graph $G_i = (V, E)$ as the *Consistency Graph* of the community C_i. Each vertex v_k in V represents an advisor community C_k from which the community C_i has requested information in the past and is described by a tuple $(Pr^+(v_k), Pr^-(v_k),\ W(v_k))$ (Table 4.1), where $Pr^+(v_k)$ represents the probability an agent a will be a *Good Contributor* in C_i given that C_k characterized it as a *Good Contributor*, $Pr^-(v_k)$ represents the probability an agent a will be a *Poor contributor* in C_i given that C_k characterized it as a *Poor contributor*, and $W(v_k)$ represents the total number of times the community C_k was asked by C_i. An example of a *Consistency Graph* is depicted in Figure 4.1(a). For instance, the community which owns the graph in Figure 4.1(a) has received information in the past from the communities $\{C_a, C_b, C_c, C_d, C_e, C_f, C_g, C_h, C_m\}$. Note that, essentially, the probability $P(v)$ over a vertex v_k depicts the degree of the consistency between the community that owns the consistency graph and the community the vertex v_k represents (i.e., C_k).

The existence of an edge $e \in E$, $e = (v_j, v_k)$, indicates that in the past the community C_i has requested information from the communities C_j and C_k regarding at least an agent a.

Each edge $e \in E$ is described by a tuple $(Pr^+(e), Pr^-(e), W(e))$ (Table 4.2), where $Pr^+(e)$ represents the probability the communities C_j and C_k will provide consistent information regarding a *Good contributor*, $Pr^-(e)$ represents the probability the communities C_j and C_k will provide consistent information regarding a *Poor contributor*, and $W(e)$ represents the degree that the amount of information that is available based on past reports is sufficient to reason about the consistency of the reports of the communities C_j and C_k. The weight $W(e)$ can be determined based on the number of times the community C_i has requested information from both the communities C_j and C_k about a common agent.

For example, the edge $e' = (v_a, v_b)$ in the Consistency Graph in Figure 4.1(a) indicates that the communities C_a and C_b were asked in the past to provide information about an agent a, and with probability $Pr^+(e') = 0.6$ and $Pr^-(e') = 0.55$ provide consistent information regarding *Good contributors* and *Poor contributors*, respectively. The latter probabilities are based on past experience and get updated each time an interaction, that involves the communities the vertices and/or the edges are associated with, takes place.

At this point we should clarify what we mean by two or more communities being *consistent*. We say that two communities have provided *consistent information* if they have both agreed on the classification of an agent (i.e., that it is either *Good* or *Poor*), and that this classification also agrees with the judgement of the recipient community.

As we will see shortly, we use the consistency graph to identify the *set* of communities that each community $C' \in S(a_j)$ appears to be consistent with in providing information and which also appear to be consistent with each other. Then we select the community with the *strongest set*, and finally from this set we select the *best* community.[1] The latter community is the first advisor community to ask. Then we remove it from the consistency graph and we repeat the procedure until we reach the number of the advisors the recipient community is interested in asking. The number of the latter communities is determined by the recipient community.

In the rest of the section we provide some necessary definitions that we use in the selection procedure. First we remind some fundamental definitions from *Graph Theory* [12] and *Game Theory* [40] and then we present our definitions.

Definition 1. *Given a graph $G = (V, E)$ the number of vertices $|V|$ is called the order of the graph and the number of the edges $|E|$ is called the size of the graph.*

Definition 2. *A path from a vertex u to a vertex u' in a graph $G = (V, E)$ is a sequence $\langle u_0, u_1, ..., u_k \rangle$ such that $u = u_0$ and $u' = u_k$ and $(u_{i-1}, u_i) \in E$ for $i = 1, ..., k$. If all vertices in the path are distinct the path is called a* simple path*.*

[1] Our definition of *strongest set* and *best community* will become clear in section 4.2.

The number of edges in the path is called the *length* of the path. For example, in Figure 4.1(a) one of the paths between the nodes v_a and v_i is $\{v_a, v_e, v_i\}$ and its length is 2.

Definition 3. *A clique in an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ of vertices, each pair of which is connected by an edge in E.*

For example, in Figure 4.1(a) the nodes v_a, v_b, v_e and v_f form a clique. The clique of the largest possible size in a given graph is called *maximum clique*.

Definition 4. *We say that a graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. Given a set $V' \subseteq V$, the subgraph of G induced by V' is the graph $G' = (V', E')$ where $E' = \{(u, v) \in E : \forall (u, v) \in E\}$*

Definition 5. *Two graphs $G = (V, E)$ and $G' = (V', E')$ are isomorphic if there exists a bijection $f : V \rightarrow V'$ such that $(u, v) \in E$ if and only if $(f(u), f(v)) \in E'$.*

For example, in Figure 4.1(b) the induced subgraph of the nodes v_a, v_b, and v_f is isomorphic to the induced subgraph of the nodes v_b, v_c, and v_d.

One very important concept of game theory that we will be using is the Bayes Nash Equilibrium. Before we formally define what a Bayes Nash Equilibrium is, we present some necessary definitions.

Definition 6. *A strategy profile $s = (s_1, ..., s_{|A|})$ is a vector that specifies the strategy s_i for each agent i in A.*

Definition 7. *Let $A = (A_1, ..., A_n)$ be the set of actions for each agent, $\Theta = \theta_1 \times ... \times \theta_n$ where θ_i is the type space of each agent where the type of an agent determines its preferences, $s_i(\theta_i)$ the strategy of each agent A_i that specifies what action (or what distribution of actions) to take for each type. A strategy profile s^* is a Bayes Nash equilibrium if $\forall i$, $\forall \theta_i$*

$$EU(s_i^*, s_{-i}^* | \theta_i) \geq EU(s_i', s_{-i}^* | \theta_i) \ \forall s_i' \neq s_i^* \tag{4.3}$$

where $EU(s_i^*, s_{-i}^* | \theta_i)$ is the expected utility of the agent A_i given its strategy s_i, its type θ_i and the strategies in s_{-i} played by the agents in $A - \{A_i\}$.

In the rest of the section we provide our proposed definitions. The first definition aims to compare a graph $G(V, E)$ with another graph $G'(V', E')$ of equal or smaller order.

Definition 8. *Given two graphs $G(V, E)$ and $G'(V', E')$, and $Pr(V)$, $Pr(V')$, $Pr(E)$ and $Pr(E')$ probability distributions over V, V', E and E', respectively, the graph $G(V, E)$*

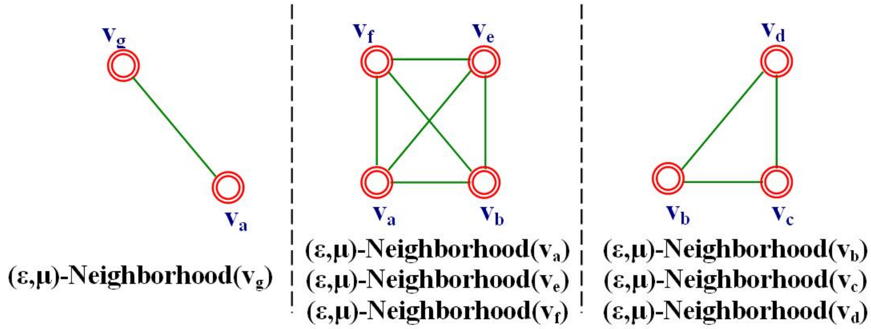

v_g
v_a
(ε,μ)-Neighborhood(v_g)
v_f
v_e
v_a
v_b
(ε,μ)-Neighborhood(v_a)
(ε,μ)-Neighborhood(v_e)
(ε,μ)-Neighborhood(v_f)
v_d
v_b
v_c
(ε,μ)-Neighborhood(v_b)
(ε,μ)-Neighborhood(v_c)
(ε,μ)-Neighborhood(v_d)

Figure 4.2: (ϵ, μ)-*Neighborhoods* for $(\epsilon, \mu) = (0.02, 0)$

(ϵ, μ)-dominates *the graph $G'(V', E')$ if there is at least one induced subgraph $G_{sb}(V_{sb}, E_{sb})$ of $G(V, E)$ isomorphic to $G'(V', E')$ that satisfies the following two conditions:*

$$mean(Pr(V_{sb})) + \epsilon \geq mean(Pr(V')) \tag{4.4}$$

and

$$mean(Pr(E_{sb})) + \mu \geq mean(Pr(E')) \tag{4.5}$$

where $\epsilon, \mu \in R^+$. In case $(\epsilon, \mu) = (0, 0)$ we will say that $G(V, E)$ strongly-dominates *the graph $G'(V', E')$.*

This definition determines whether the information a graph $G'(V', E')$ encodes is equal to, or richer than, the information a graph $G(V, E)$ of equal or larger order.

Proposition 1. *A graph* strongly dominates *all of its induced subgraphs.*

Proof. A graph $G(V, E)$ *strongly dominates* a graph $Q(V_Q,\ E_Q)$ if it has a subgraph $G'(V', E')$ isomorphic to Q such that:

$$mean(Pr(V_{G'})) \geq mean(Pr(V_Q))$$

and

$$mean(Pr(E_{G'})) \geq mean(Pr(E_Q))$$

This is true since given that Q is an induced subgraph of G, we can simply select for G' the graph Q itself. □

Definition 9. *Given a clique $Q(V_Q, E_Q)$ of a graph $G(V, E)$, the* set of external edges *of the clique $Q(V_Q, E_Q)$ is the set of all edges $e = (v, w)$ in E such that $v \in V_Q$ and $w \in V - V_Q$.*

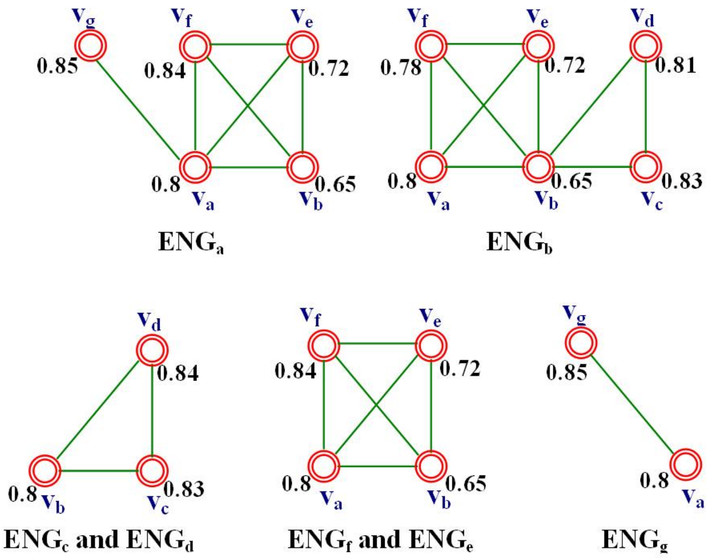


Figure 4.3: Extended Neighborhood Graphs

Definition 10. *Let $G(V, E)$ be a graph with $Pr(V)$ and $Pr(E)$ probability distributions over the set of its vertices V and the set of its edges E, respectively, and let S be the set of the maximal cliques that the vertex $v \in V$ participates in. The* (ϵ, μ)-Neighborhood *of v is the clique in S that:*

- (ϵ, μ)-dominates *all the other cliques of smaller order that the vertex v participates in, and*
- *strongly dominates all the other cliques of equal order.*

In the case where there is more than one clique that satisfies the above requirement, the (ϵ, μ)-Neighborhood(v) *is the clique with the largest* set of external edges.

We define the (ϵ, μ)-*Neighborhood* of a node v based on both the maximal cliques but also on the (ϵ, μ)*-domination* criterion because we are interested in finding the clique in which v participates in that combines both the order and the quality of the information the clique encodes. As we will see in Section 4.2.4 the (ϵ, μ)-*Neighborhood(v_i)* graph consists of communities that provide consistent information with the community C_i and at the same time provide consistent information with each other. We argue that asking one community in this graph provides approximately the same value of information as asking any other community in it. An example of (ϵ, μ)-*Neighborhoods* is depicted in Figure 4.2. These (ϵ, μ)-*Neighborhoods* are constructed based on the Consistency Subgraph in Figure 4.1(b) for $(\epsilon, \mu) = (0, 0.02)$.

The next step is to select one of the (ϵ, μ)-*Neighborhoods*. While we could merely select the graph with largest order in the (ϵ, μ)-*Neighborhood*, we note that this would ignore some interesting features of the definition of these neighborhoods. For instance, note that while the neighborhoods of v_a, v_e and v_f in Figure 4.2 are the largest, the nodes v_a and v_b are also members of the neighborhoods of v_g and v_c, respectively. This means that v_a and v_b are consistent with more communities than v_e and v_f and, thus, they capture richer information.

We introduce the (ϵ, μ)-*Extended Neighborhood Graph*, ENG_i, of each node v_i as the union of v_i's (ϵ, μ)-*Neighborhood* with all the other (ϵ, μ)-*Neighborhoods* in which v_i is a member.

Definition 11. *The* (ϵ, μ) *Extended Neighborhood Graph of a node* v_i*,* ENG_i*, is the union of all the* (ϵ, μ)-Neighborhoods *in which the node* v_i *is a member.*

We refer to v_i as the *owner node* of ENG_i while we refer to the rest of the nodes in ENG_i as *peripheral nodes* while for ease of presentation we will refer to the (ϵ, μ) Extended Neighborhood Graph as the Extended Neighborhood Graph. The *ENG*s of the nodes v_a, v_b, v_c, v_d, v_e, v_f, v_g of the (ϵ, μ)-*Neighborhoods* in Figure 4.2 are depicted in Figure 4.3.

Definition 12. *The* Dominant ENG *of a set* S *of ENGs is the graph in* S *with the maximum order that* (ϵ, μ)*-dominates all the graphs in* S *of smaller or equal order.*

4.2.2 The Advisor Selection Algorithm

In this section we describe the *Advisor Selection Algorithm* for finding a set of advisors L. Although our main aim is to identify the communities which provide consistent and accurate information regarding agents who are *Good contributors*, we are also interested in asking a small number of communities that provide consistent information about agents who are *Poor contributors*. This is due to the fact that communities might be interested in misreporting only good contributors in fear of losing them or misreporting only poor contributors in an effort to get rid of them.

The list L^+ consists of the communities that tend to provide consistent information about agents who are *Good contributors* while the list L^- consists of the communities that tend to provide accurate information about agents who are *Poor contributors*. Obviously, $L = L^+ \cup L^-$. Briefly, the procedure of finding list L^+ or L^- is as follows:

- Filter the consistency graph by removing the vertices and edges from the consistency graph which are of no value. We will refer to the graph that occurs as the *consistency subgraph*.

- Find the (ϵ, μ)-*Neighborhoods* for each vertex in the consistency subgraph *(goal: Identify the strongest neighborhood each vertex participates in.)*
- Find the *Extended Neighborhood Graph* for each vertex in the consistency subgraph. *(goal: Identify the extended neighborhood the vertex participates in.)*
- Find the *Dominant Extended Neighborhood Graph. (goal: Identify the strongest extended neighborhood.)*
- Select the winner node w of the Dominant Extended Neighborhood Graph. *(goal: Identify the strongest node in the strongest extended neighborhood.)*
- Insert the community the node w represents in the list of the advisors.
- Remove w from the consistency graph and repeat the procedure.

In the next sections we focus on finding the list L^+ and thus we will only consider the probability distributions $Pr^+(V)$ and $Pr^+(E)$. The list L^- can be computed in exactly the same way by simply replacing the probability distributions $Pr^+(V)$ and $Pr^+(E)$, with $Pr^-(V)$ and $Pr^-(E)$, respectively.

4.2.3 Filtering the Consistency Graph

The filtering step in the *Selection Procedure* refers to removing the nodes that represent communities that either the community C_i would like to ask anyways or represent communities that provide insufficient information. For example, if a community provides accurate information with probability 0.1 then this community would be a bad choice, thus it should be removed from the candidates list.

We refer to the graph that occurs from the consistency graph when removing nodes as the *consistency subgraph* and we represent it by $Q_i(V_{Q_i}, E_{Q_i})$. In particular, the *consistency subgraph* Q_i is created by removing:

1. the *vertices* which represent communities that do not belong to $S(a_j)$, the set of communities the agent a_j has been a member, or represent communities whose probability of telling the truth regarding *Good contributors* is less than an acceptable threshold θ_v^+, or represent communities for which there is insufficient experience and thus their $W(v)$ is less than an acceptable threshold θ_v^{ep}, and

2. the *edges* which connect communities whose probability of agreeing about *Good contributors* is lower than an acceptable threshold θ_e^+, or edges for which there is insufficient experience and thus their $W(e)$ is less than an acceptable threshold θ_e^{ep}.[2]

[2] The thresholds θ_v^{ep} and θ_e^{ep} are set by each community and can be computed based on the community's experience.

An example of a *consistency subgraph* is depicted in Figure 4.1(b). More specifically, the latter graph is created from the consistency graph G_i in Figure 4.1(a) if we consider that the candidate advisor list is $S(a_j)$={a,b,c,d,e,f,g,h}, $\theta_v^+ = 0.65$, $\theta_v^{ep} = 0.6$, $\theta_e^{ep} = 10$, and $\theta_e^+ = 10$. For example vertex i is removed since it does not belong in $S(a_j)$, while the vertex h and the edge $e_1 = (v_e, v_g)$ are removed because there is insufficient previous experience (i.e., $W(h) < 10$ and $W(e_1) < 10$). At this point we have to clarify that in order to accumulate experience for the above nodes, each time a number of *unexplored* nodes can be selected to be asked with some probability p.

4.2.4 Finding the Dominant Extended Neighborhood Graph

The next step is to exploit the consistency in advice that the communities in the *consistency subgraph* provide. In order to achieve this we need to identify the *Dominant **E**xtended **N**eighborhood **G**raph.* The *Dominant ENG* is a graph that contains the set of communities that tend to provide the best quality of information while at the same time are consistent with each other.

First we need to find the (ϵ, μ)-*Neighborhood* of each community in the consistency subgraph. The (ϵ, μ)-*Neighborhood* of a community C_j consists of the set of communities that are consistent not only with the community C_j but with each other as well. Our goal is to identify the set of communities that C_j belongs to and which has the following property: asking C_j is equivalent to asking any of the communities in the latter set, since if everybody tends to agree with everybody else in the set then simply asking one of them is sufficient.

The next step is to construct the *Extended Neighborhood Graph*, ENG_j, of each node v_j in the consistency subgraph. Recall, we refer to the node v_j as the *owner node* of the graph ENG_{C_j}. As we described in *Definition 4* this is done by merging all the (ϵ, μ)-*Neighborhoods* the community C_j participates in. If a community C_j participates in C_a's (ϵ, μ)-*Neighborhoods* then asking C_j or C_a's is equivalent. Thus, if the community C_j also participates in C_b's (ϵ, μ)-*Neighborhood* then asking C_b is equivalent to asking C_j. For example, consider the (ϵ, μ)-*Neighborhoods* in Figure 4.2. The node v_a participates in v_g's (ϵ, μ)-*Neighborhoods* thus, its *Extended Neighborhood Graph* is created by merging its (ϵ, μ)-*Neighborhood* with node's v_g (ϵ, μ)-*Neighborhood.*

The final step is to find the *Dominant Extended Neighborhood Graph.* For a similar reason to the one that led us to consider the (ϵ, μ)-*Neighborhoods* instead of only considering the maximum clique a node belongs to, we find the *Dominant Extended Neighborhood Graph* by applying the (ϵ, μ)-domination condition on the set of *Extended Neighborhood Graphs.* More specifically, the *Dominant ENG* is the largest *Extended Neighborhood Graph* that (ϵ, μ)-*dominates* all the *Extended Neighborhood Graphs* of equal or smaller order.

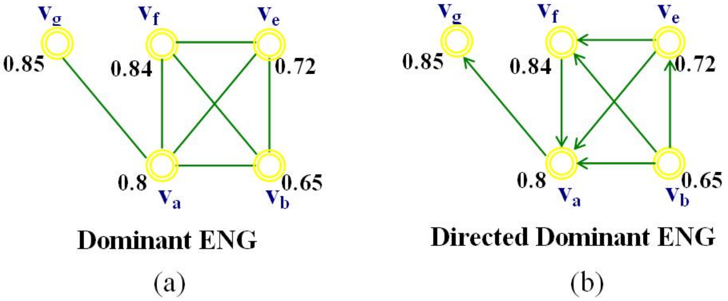


Figure 4.4: Directed ENG

4.2.5 Finding the Winning Advisor

The next step is to determine which community that is represented by a node in the *Dominant Extended Neighborhood Graph* to ask. We will refer to the latter node as the *winning node* of the *Dominant Extended Neighborhood Graph* and the community it represents as the *winning advisor*.

Towards this goal, we turn the *Dominant Extended Neighborhood Graph* into a directed graph. In particular, for each edge $e = (v, w)$ of the graph we give a direction from w to v if $Pr(v) > Pr(w)$, from v to w if $Pr(v) < Pr(w)$ and a double direction if $Pr(v) = Pr(w)$, where Pr is the probability distribution over the vertices. Then we simply choose the node with the highest indegree.[3] For example, assume that the *Dominant Extended Neighborhood Graph* among the graphs in Figure 4.3 is the graph ENG_a in Figure 4.4(a). By adding direction on its edges the graph in Figure 4.4(b) is created. As we can see the node with the maximum indegree is v_a. Note, our approach guarantees that if the node with the highest probability is the owner of the *Dominant Extended Neighborhood Graph* it will be selected since there is no peripheral node with a degree higher than the owner's degree.

Summarizing, the algorithm for finding the ordered list L of the advisors is:

[3] Tie breaking order: highest probability, owner node, number of past interactions.

```
ASA
Input: $G(V, E)$, $\epsilon$, $\mu$, $Pr^*(V)$, $Pr^*(E)$, $S$
Output $L^*$

  Find the consistency subgraph $Q(V_Q, E_Q)$ from $G(V, E)$ and $S$
  while $L^*$ not full AND $V_Q$ not empty do
    for all  $v \in V_Q$ do
      Find the $(\epsilon, \mu)$-Neighborhood(v) in $Q(V_Q, E_Q)$
    end for
    for all $v_k \in V_Q$ do
      Find the Extended Neighborhood Graph $ENG_k$
      Add $ENG_k$ in list $ST$
    end for
    Find the Dominant Extended Neighborhood Graph $ENG_p \in ST$
    for all $e = (v, w) \in E_{ENG_p}$ do
      if $Pr^*(v) < Pr^*(w)$ then
        Give edge $e$ direction $v \rightarrow w$
      else if $Pr^*(v) > Pr^*(w)$ then
        GIVE edge $e$ direction $v \leftarrow w$
      else
        Give edge $e$ directions $v \rightarrow w$ and $v \leftarrow w$
      end if
    end for
    Find the node $k \in V_{ENG_p}$ with the maximum in-degree.
    Add $k$ at the end of the list $L^*$
    Remove $k$ from $V_Q$ and all incident edges
  end while
```

4.2.6 Incentive Compatibility

In this section we prove that in our approach honesty is a Bayes Nash Equilibrium and thus our approach is incentive compatible. First, we provide some necessary propositions and then we move to our main theorem. Note, in our mechanism, we refer to a community as *honest* if it provides correct information and *dishonest* otherwise.

Proposition 2. *Let $ENG_k(V, E)$ be a* Dominant Extended Neighborhood Graph*, $Pr(V)$ and $Pr(E)$ probability distributions over V and E, and let v_k be the owner node. If the node v_k has the largest probability in V then it is always the winning node.*

Proof. Since v_k is the owner node and has the largest probability compared to any other node in $ENG_k(V, E)$ then it must be the case that v_k's indegree is equal to its degree. Since v_k participates in all maximal cliques of ENG_k then it must be the case that its degree is greater than, or equal to, the degree of any other node in V, and thus its indegree is greater than, or equal to, any node. If v_k's indegree is greater than all other nodes, then by definition, it will be selected as the winning node. If its indegree is equal to at least one other node, then by our tie-breaking rules v_k will still be selected. □

Proposition 2 shows that: a) it is in the best interest of a node to be the owner node, and b) it is in the best interest of the owner node to have and maintain the largest probability among the other nodes in its ENG.

Proposition 3. *Let $ENG_k(V, E)$ be the* Dominant Extended Neighborhood Graph*, $Pr(V)$ and $Pr(E)$ probability distributions over V and E, respectively, and v_k its owner node. If the winning node v'*

- *is a peripheral node then an update from $Pr(v_k) < Pr(v')$ to $Pr(v_k) > Pr(v')$ results in the change of the winning node*
- *is the owner node then an update from $Pr(w) < Pr(v')$ to $Pr(w) > Pr(v')$ does not always result in a change of the winning node*

Proof. Let v' be a peripheral node. By definition the owner node v_k participates in every (ϵ, μ)-*Neighborhood* in its *Extended Neighborhood Graph.* If v_k's probability becomes larger than the previous winning node's v', then v_k will have higher probability than all the nodes v' does, thus its indegree will be equal or higher to v''s indegree plus 1. Thus, the node v' cannot be the new winning node.

Let v' be the owner node (i.e., $v' = v_k$). First, note that even if a peripheral node has a higher probability than the owner node of a *Dominant ENG* graph, it is not guaranteed to be the winning node. For instance, in Figure 4.4 the node v_g has probability higher than v_a but its indegree is lower than the owner's. Thus, if w's probability becomes higher than the owner node's v_k, this does not guarantee that v' will not be the new winning node. □

Proposition 3 shows, that i) it is in the best interest of a node to be the owner node of the *Dominant ENG*, and ii) it is in the best interest of any peripheral node to have and maintain a probability higher than the owner node, and thus the highest probability in each (ϵ, μ)-*Neighborhood* it participates in.

Proposition 4. *Let $ENG_k(V, E)$ be the* Dominant Extended Neighborhood Graph*, $Pr(V)$ and $Pr(E)$ probability distributions over V and E, respectively, and w the winning node. There is no clique $Q'(V_{Q'}, E_{Q'})$ in which w does not participate in (i.e., $w \notin V_{Q'}$) such that:*

- $|V_{Q'}| > indegree(w) + 1$, *and*
- $max\{Pr(V_{Q'})\} > Pr(w)$.

Proof. Assume that there is a clique $Q'(V_{Q'}, E_{Q'})$ in ENG_k such that $w \notin V_{Q'}$, $|V_{Q'}| > indegree(w) + 1$ and $max\{Pr(V_{Q'})\} > Pr(w)$. Let q be a node in $V_{Q'}$ such that $q = argmaxPr(V_{Q'})$. From the latter we have $indegree(q) \geq indegree(w)$. Given now that q has higher probability $Pr(q)$ than w (i.e., $Pr(q) > Pr(w)$) the node w cannot be the winning node. Thus, with the existence of q then node w is no longer the winning node, consequently such clique as Q' does not exist. □

Proposition 4 shows that an update of the probability distribution of a *Dominant Extended Neighborhood* $ENG_k(V, E)$ that creates such a clique results in the replacement of the winning node w by another node.

Proposition 5. *Let* $ENG_k(V, E)$ *be the* Dominant Extended Neighborhood Graph, $Pr(V)$ *and* $Pr(E)$ *probability distributions over* V *and* E*, respectively, and* v' *is the winning node in* V*. Let* O *be the descending order of the nodes in* V *based on the probability distribution* $Pr(V)$*. Any update on* $Pr(V)$ *that*

- *does not influence the order of the nodes* V-$\{v'\}$ *in* O*, and*
- *increases or maintains the ranking of the node* v' *in* O

does not influence the selection of the winning node.[4]

Proof. If the order of the nodes with respect to $Pr(V)$ does not change then the indegree of each node which is not adjacent to v' will remain the same, while the indegree of nodes that are the adjacent to v' will either decrease or remain the same. Thus, v' will remain the winning node.[5] □

Proposition 5 states that a winning node is better off maintaining or increasing its probability.

Now, we would like to prove that honest reporting is a Bayes Nash Equilibrium. We will show if all agents in $A - \{a_i\}$ are honest in their reporting, then agent a_i is also best off, in expectation, when it reports honestly.

[4] Note that the nodes are ordered in descending order based on their probability, thus the node with 0 has the highest probability.

[5] If the relation between the probabilities of the vertices does not change the direction of the edge will also remain the same.

Theorem 1. *If all other communitiess are honest, a community C_i by being dishonest can only reduce the likelihood of being selected as a winning node, and thus decreases the probability of being asked.*

Proof. Given that the expected utility of a community C_i is

$$EU(i) = Pt^i_{provide} \cdot Pr^i_{provide} - Pt^i_{request} \cdot Pr^i_{request} \tag{4.6}$$

and that C_i has no control over the probability $Pr^i_{request}$, it is sufficient to show that when C_i is honest, $Pr^i_{provide}$ either does not change or increases. The community C_i does not have a direct control over the probability $Pr^i_{request}$ since it cannot control whether other agents will be interested to join it. More specifically, the community C_i does not have control over the upper bound of $Pr^i_{request}$

Assume, without loss of generality, that community C_1 had requested information from a set L of communities, with $C_i \in L$. For any $C_j \in L$ let v_j denote the node representing the community C_j in the consistency graph of C_1, which we denote by $G_1 = (V_1, E_1)$. Assume also, that all agents in $L \setminus \{C_i\}$ reported honest information to C_1, whereas C_i lied. We will show that this lie will only decrease the likelihood that C_i will be selected to provide information in the future (i.e., $Pr^i_{provide}$ will decrease). Since all $C_j \in L \setminus \{C_i\}$ were honest, then they were also consistent with each other, whereas, C_i lied and thus was inconsistent. Therefore, G_1 would be updated such that $Pr(v_j)$ would increase (or remain the same if $Pr(v_j) = 1$), while $Pr(v_i)$ would decrease. Similarly, for any edge $e = (v_j, v_k) \in E_1$ such that $j, k \neq i$, $Pr(e)$ would increase (or remain the same if $Pr(e) = 1$). However, for any edge $e' = (v_i, v_j)$ the probability assigned to the edge would remain the same. Let G_1^* denote the new consistency graph of C_1 after these updates have been made.

Given the new consistency graph of C_1, G_1^*, if the community C_1 were to want information again, it would only select community C_i if the vertex v_i was a member of the dominant extended neighborhood graph of G_1^*. We will now analyze the scenarios that led v_i to be selected in G_1 and how this is related to its current chance of being selected in G_1^*. Let ENG_k be any dominant extended neighborhood graph of G_1 that v_i might have participated in. There are two possible scenarios: i) v_i was the owner node, or ii) v_i was a peripheral node. We now investigate these cases.

Case 1: v_i was the owner node of ENG_k. Assume that node v_i was selected and was the node with the highest probability in ENG_k (Proposition 2). In G_1^*, all nodes $v_j \neq v_i$ have had either their probability increase (or remain the same), while in G_1^* the probability of v_i has decreased. Therefore, by misreporting its information when asked, v_i decreases the likelihood of continuing being the node with the largest probability, and thus decreases the probability of being the winning node since it is only guaranteed that the owner node will be always the winning node if it has the largest probability. Assume that v_i was selected as the winning node but did not have the highest probability in G_1. Considering the Propositions 4 & 5, in order for v_i to be the winning node in ENG_k, given that all the other nodes increase or maintain their probabilities in G_1^*, v_i has also to maintain or increase its probability $Pr(v_i)$ (i.e., by telling the truth). Thus, by lying C_i reduces the possibility of v_i being the winning node.

Case 2: v_i was a peripheral node in ENG_k. Since v_i is a peripheral node in G_1 and is also a winning node in G_1, it must be the case that the owner node, v_k, of ENG_k was such that $Pr(v_k) < Pr(v_i)$. In G_1^*, we have that $Pr(v_i)$ decreases while $Pr(v_k)$ increases or remains 1. Therefore, in G_1^* either v_i will remain a winning node (as long as $Pr(v_k) > Pr(v_i)$) or will no longer be a winning node (if $Pr(v_k) > Pr(v_i)$) (Proposition 3). Therefore, by lying, either the community C_i does not change its status as a winning node, or else becomes a losing node, and is thus not selected.

Furthermore, by following a similar methodology as above, it can be showed that when C_i lies the probability that v_i will be the owner node of a *Dominant ENG* and the probability that it will participate in the *Dominant ENG* decreases. The reason is that node v_i by decreasing the probabilities that are associated with it weakens all the (ϵ, μ)-*Neighborhoods* it could participate in. Given now that every community in $L \setminus \{C_i\}$ increases its probability it strengthens the (ϵ, μ)-*Neighborhoods* it participates in. The probability the *Extended Neighborhoods* v_i participates in will (ϵ, μ)-*dominate* other *Extended Neighborhoods* decreases. □

4.3 Summary

Summarizing, in this chapter we focused on providing incentives to communities to truthfully exchange information regarding their member agents. In particular, we consider that a community contacts the communities (advisors) the agent is or was previously a member and requests information about the agent. We depict the consistency of the advice the advisors provide in a graph, the consistency graph, and we select a set of induced subgraphs which we use to build the subset of advisors the community should consult. We show that our approach leads to a competitive information environment which provides appropriate incentives for the advisors to provide truthful reports. In particular, we presented an algorithm for selecting a subset of advisors by exploiting the consistency in providing honest advice among a set of advisor communities and we showed that honesty is a Bayes Nash Equilibrium. In our mechanism, we assume that the *honest* communities can accurately reason about the behaviour of their agents. Even though this appears to be a strong assumption, this is not necessarily the case since each community is self-interested and thus it would like to collaborate with communities that provide information that it can use.

Chapter 5

The Payment Decision Problem

In this chapter we present a scoring function that is used to determine the payment each advisor community will receive. In particular, as we will discuss, the requirement for the payment function is to be monotonically increasing with respect to the value of the scoring function. We refer to the scoring function as the *i-Function* and to its value as the *i-Score*. Our goal is the use of an *i-Function* that promotes honest exchange of information regarding the evaluation of an agent. Our scoring function is motivated by the work on scoring rules [67]: a framework for eliciting probabilistic information from agents.

The two main issues we are interested in addressing are:

1. how an advisor community can be motivated to truthfully report its ratings and
2. how we can value the quality of the rating a community provides in order to compensate it with a fair payment.

As will be seen, we set our *i-Function* to be maximized only when the community provides a truthful rating (for 1.) and we introduce a set of properties the scoring function should follow in order to promote honesty and fairness (for 2.), thus, providing an effective proposal for the exchange of evaluation information between communities.

A key distinction that we make is between what we refer to as a *good* or a *poor* contributor. These are labels that correspond to the desirability for a community to accept or to reject the agent (the key decision-making that each community must undergo). The evaluation information that is shared consists of both a rating which can represent either the reputation or the trustworthiness of the agent and a type which can have either the value *good* and *poor* which is the interpretation of rating. The community receiving the agent's evaluation reason about the accuracy and the helpfulness of this information. As we will show, evaluation information that leads to the correct decision concerning the acceptance or rejection of an agent leads to more lucrative payments, thus promoting both honest reporting and fair payments.

In the following sections, we provide an overview of our model, then introduce the properties that we believe should be taken into consideration for determining the *i-Score* each community should receive. Finally, we provide an example of a family of scoring functions that satisfies these properties.

5.1 Model

Let C_i denote community i and let a_j denote agent j. We assume that if a_j is a member of community C_i then C_i can observe and judge the quality of agent a_j. In particular, we assume that community C_i maintains an *evaluation model* for all member agents, and is able to assign an *evaluation rating* r^i_j to agent a_j, where r^i_j is some real number from the interval $[\alpha, \beta]$, $0 \leq \alpha < \beta$.

If agent a_j wishes to join community C_i, then before welcoming a_j, community C_i will contact a set of communities in which a_j is currently, or was previously, a member. We denote the set of these communities as $m(a_j)$. Recall, the set $m(a_j)$ is determined by our *Advisor Selection Algorithm.* The communities in $m(a_j)$ are asked to provide two pieces of information. First, each community $C_k \in m(a_j)$ is asked to report its evaluation rating r^k_j for agent a_j. Since communities may use different evaluation models, or may interpret ratings differently, communities are also requested to provide *type information*, θ^k_j, for agent j where $\theta^k_j \in \{\text{good}, \text{poor}\}$. The type θ^k_j is essentially the *interpretation* that community C_k makes concerning its evaluation rating r^k_j for agent a_j.[1]

In exchange for information, community C_k receives a payment from community C_i. This payment, P, depends on how useful or *important* community C_i finds the information provided by C_k. If community C_i is interested in possibly welcoming agent a_j, it will contact the communities in $m(a_j)$ to request information about the agent a_j. Each community $C_k \in m(a_j)$ reports its evaluation rating (r^k_j) and type information (θ^k_j) for agent a_j, possibly misreporting the information.

Based on the information received from communities in $m(a_j)$, C_i decides whether to accept agent a_j or to reject it. If C_i accepts agent a_j then it gets to *observe* and *evaluate* a_j. By doing so, C_i is able to assign both a rating, $\hat{r}$, and a type, $\hat{\theta}$ to the agent. If community C_i decides to *reject* agent a_j, then the payment procedure is a little more complicated

[1] For example, a rating 0.55 in one community could indicate an agent which is a poor contributor while in another community, which might be more strict in providing high ratings, the same rating might indicate a good contributor.

since the community never gets the opportunity to directly observe and evaluate agent a_j. Instead, community C_i sets the type of agent a_j to be $\hat{\theta} = \text{poor}$, and computes $\hat{r}$ to be the average evaluation rating of all communities in $m(a_j)$ that also assigned type equal to *poor*. That is, if $B = \{C_k | C_k \in m(a_j) \text{ and } \theta_j^k = \text{poor}\}$ then:

$$\hat{r} = \frac{\sum_{C_k \in B} r_j^k}{|B|}.$$

Only after this decision is made do the communities in $m(a_j)$ get paid for the information that they provided. In particular, for each community $C_k \in m(a_j)$, the payment it receives for its information (r_j^k and θ_j^k) is determined by the payment function,

$$P(I(r_j^k, \hat{r}, \theta_j^k)) : \mathbb{R}^+ \mapsto \mathbb{R}^+ \tag{5.1}$$

where r_j^k is the evaluation rating of the agent by a community $C_k \in m(a_j)$, $\hat{r}$ is the evaluation rating of the agent by the recipient community, $I(x, \hat{r}, \theta)$ is the *i-Function* used by community C_k to determine the *importance* of the information provided by community C_k. The main requirements for the payment function is for it to be a monotonically increasing function with respect to the *i-Score* $I(r_j^k, \hat{r}, \theta_j^k)$. Thus, the key part of the payment function is the *i-Function*.

In the following sections we present the properties we believe the *i-Function* should exhibit, a family of functions that satisfy these properties and two particular instances.

5.2 The *i-Function*

In this section we describe the basic desirable properties of the *i-Function*. In the previous section we informally introduced *i-Function* as $I(r_j^k, \hat{r}, \theta_j^k)$ where r_j^k was the evaluation rating for agent a_j given by community C_k, $\hat{r}$ was community C_i's (the community making the payment) evaluation rating of the agent, and θ_j^k was C_k's assigned type to the agent. We define I as $I : \mathbb{R} \times \mathbb{R} \times \{\text{good}, \text{poor}\} \to \mathbb{R}$ such that $I(x, \hat{r}, \theta) = -\infty$ if $x < \alpha$ or $x > \beta$ for some predefined $\alpha, \beta \geq 0$. Because of this, a community C_k, reporting on agent a_j is best off revealing a *legal* rating. We will refer to the outcome of the *i-Function* as *i-Score*.

Our first desired property is that I is continuous when a legal evaluation rating is given.

Property 1. *Let* $\alpha \geq 0$. *In the restricted domain* $[\alpha, \beta] \times [\alpha, \beta] \times \{\text{good}, \text{poor}\}$, *$I$ is continuous.*

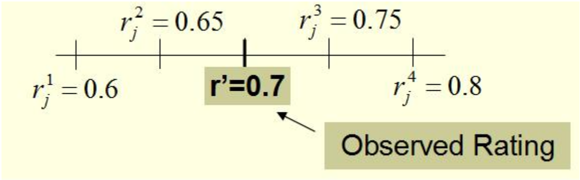


Figure 5.1: Example of ratings

If the community accepts the agent, then it is able to observe and evaluate the agent, and thus determine its own evaluation rating, $\hat{r}$. When determining payments we desire that the communities that provided the *more accurate* information are rewarded. Consider the following example.

Example 1. *Assume that community C_i requests information about agent a_j. Assume that four communities submitted evaluation ratings $r_j^1 = 0.6$, $r_j^2 = 0.65$, $r_j^3 = 0.75$ and $r_j^4 = 0.8$. After observing the agent, community C_i set $\hat{r} = 0.7$ (Figure 5.1).*

We would like our *i-Function* to reward the communities that submitted evaluation ratings of 0.65 and 0.75 more than those that submitted 0.6 and 0.8, respectively. That is, we would like to capture the property that as x approaches $\hat{r}$, the importance of x increases. Property 2 captures this.

Property 2. *For any $\hat{r} \in [\alpha, \beta]$ and for any $\theta \in \{\text{good}, \text{poor}\}$, if $x \in [\alpha, \hat{r}]$ then $I(x, \hat{r}, \theta)$ is strictly monotonically increasing, and if $x \in [\hat{r}, \beta]$ then $I(x, \hat{r}, \theta)$ is strictly monotonically decreasing.*

Given Property 2, the *i-Function* should reward communities C_2 and C_3 in Example 1 more than communities C_1 and C_4, respectively. We may want, however, to further distinguish between the information received from communities C_2 and C_3 since even though the evaluation values were equally far from $\hat{r}$, C_2 stated that a_j had a lower evaluation than observed, while community C_3 stated a_j had a higher evaluation than observed.

Define

$$\delta(\hat{r}, \epsilon, \theta) = I(\hat{r} + \epsilon, \hat{r}, \theta) - I(\hat{r} - \epsilon, \hat{r}, \theta)$$

for any $\epsilon \in (0, \min[\hat{r} - \alpha, \beta - \hat{r}]]$. This measures the difference in the *i-Score* when communities over-report and under-report by the same amount. If $\delta(\hat{r}, \epsilon, \theta) = 0$ for all ϵ then the *i-Function* would treat over- and under-reported evaluation ratings equally. We believe that the *i-Function* should be used to reward communities that provide ratings which deviate towards the *correct direction* (i.e., good or poor) higher than those communities who provided ratings of equal deviation but towards the wrong direction.

Property 3. *For any $\hat{r}$ and any $\epsilon \in (0, \min[\hat{r} - \alpha, \beta - \hat{r}]]$ let $I(x, \hat{r}, \theta)$ be such that*

$$\delta(\hat{r}, \epsilon, \text{good}) > 0$$

and

$$\delta(\hat{r}, \epsilon, \text{poor}) < 0.$$

In words, Property 3 says that if C_i determines that the agent is a good agent, then communities who reported higher evaluation ratings should be rewarded more than communities who reported lower evaluation ratings, assuming that the difference from $\hat{r}$ is the same. A similar property should hold if C_i determined that the agent was poor. Referring back to Example 1, if C_i determined that $\theta = \text{good}$, then community C_3 should have a higher *i-Score* than C_2, and C_4 should have a higher *i-Score* than C_1. If $\theta = \text{poor}$, then C_2 should have a higher *i-Score* than C_3 and C_1 should have a higher *i-Score* than C_4.

Our last two desired properties describe how $\delta(\hat{r}, \epsilon, \theta)$ should behave.

Property 4. *For any $\hat{r}$*

- $\delta(\hat{r}, \epsilon, \text{good})$ *is strictly monotonically increasing in ϵ,*
- $\delta(\hat{r}, \epsilon, \text{poor})$ *is strictly monotonically decreasing in ϵ.*

We interpret Property 4 in that if an agent is judged to be good, then communities who submitted high evaluation ratings for the agent deserve higher *i-Score*, since they were offering support for the agent (and vice-versa for the case when an agent is judged to be poor). Referring back to Example 1, if the agent was judged to be good, then community C_4 would receive the highest *i-Score*, whereas if the agent was judged to be poor, then community C_1 would receive the highest *i-Score*.

Property 5. *For any $\epsilon \in (0, \min[\hat{r} - \alpha, \beta - \hat{r}]]$ and for any $\theta \in \{poor, good\}$, $\delta(\hat{r}, \epsilon, \theta)$ is monotonically decreasing in $\hat{r}$.*

Property 5 states that a given deviation ϵ has different significance for different values of $\hat{r}$. For instance, if the agent's type is judged to be good then the significance of a deviation ϵ increases as the reported rating $\hat{r}$ decreases. This is due to the fact that as $\hat{r}$ decreases the rating $\hat{r} - \epsilon$ might be crossing the cutoff $\bar{r}$ value that a community considers in order to accept an agent or not. In particular, the further the rating $\hat{r} - \epsilon$ crosses $\bar{r}$ the more in doubt it can put a community regarding the agent's real value. Analogous, is the case where the agent's type is judged to be poor.

5.2.1 *i-Function* and Payments

The *i-Function* forms the foundation of our payment system, and thus the properties of the *i-Function* have a profound influence on the properties of the payments, and the incentives for communities to report their evaluation ratings and type information when requested.

We first note that if the *i-Function* satisfies Properties 1 and 2, then it is uniquely maximized when communities report $\hat{r}$ (*i.e.,* $I(\hat{r}, \hat{r}, \theta)$ is the global maximum). Since the payment a community receives is monotonically increasing with respect to the *i-Scores*, a community has incentive to report the evaluation rating that it truly believes C_i will experience if C_i accepts the agent since this will result in the highest possible *i-Score.* We introduce Properties 3 through 5 so as to ensure a certain level of *fairness* in the system. While the communities who present the most accurate information to community C_i benefit the most from the *i-Function*, communities, which provide information that tried to convince community C_i to make the appropriate decision with respect to the agent, are also well rewarded payment-wise.

5.3 A Class of *i-Functions*

In the previous section we outlined the desirable properties for an *i-Function.* The obvious question is then *"Does there exist any functions which could be used as* i-Functions*?"* In this section we introduce a class of functions that satisfy Properties 1 to 4, and that contains a subclass which satisfies Property 5.

Let $\phi : [\alpha, \beta] \to \mathbb{R}^+$ and $\psi : [\alpha, \beta] \to \mathbb{R}^+$ be arbitrary continuous functions on $[\alpha, \beta]$. Let ϕ be strictly monotonically decreasing and let ψ by strictly monotonically increasing. Now, define the *i-Function* $I(x, \hat{r}, \theta)$ as

$$I(x, \hat{r}, \theta) = \begin{cases} \int_{\alpha}^{\beta} \phi(y)dy - |\int_{\hat{r}}^{x} \phi(y)dy|, & \text{if } \theta = \text{good} \\ \\ \int_{\alpha}^{\beta} \psi(y)dy - |\int_{\hat{r}}^{x} \psi(y)dy|, & \text{if } \theta = \text{poor} \end{cases} \tag{5.2}$$

We now show that for any choice of ψ and ϕ, $I(x, \hat{r}, \theta)$ satisfies Properties 1 to 4. Later we show that for particular choices of ϕ and ψ it is possible to also satisfy Property 5.

First, since both ϕ and ψ are continuous on $[\alpha, \beta]$, then $I(x, \hat{r}, \theta)$ is also continuous on the restricted domain $[\alpha, \beta] \times [\alpha, \beta] \times \{\text{good}, \text{poor}\}$. That is, $I(x, \hat{r}, \theta)$ satisfies Property 1. As the x approaches $\hat{r}$ from the left the area that is defined by $|\int_{x}^{\hat{r}} \phi(y)dy|$ is strictly decreasing. Consequently, $I(x, \hat{r}, \theta)$ is strictly increasing in $[a, \hat{r}]$. As the x goes away from $\hat{r}$ then the area that is defined by the $|\int_{\hat{r}}^{x} \phi(y)dy|$ is strictly increasing. Thus, $I(x, \hat{r}, \theta)$ is strictly decreasing in $[\hat{r}, \beta]$. Similarly, it can be proved that the function ψ satisfies Property 2 as well. Property 3 is also satisfied. We can rewrite $\delta(\hat{r}, \epsilon, \theta)$ as:

$$\delta(x, \hat{r}, \theta) = \begin{cases} -\int_{\hat{r}}^{\hat{r}+\epsilon} \phi(y)dy + \int_{\hat{r}-\epsilon}^{\hat{r}} \phi(y)dy, & \text{if } \theta = \text{good} \\ \\ -\int_{\hat{r}}^{\hat{r}+\epsilon} \psi(y)dy + \int_{\hat{r}-\epsilon}^{\hat{r}} \psi(y)dy, & \text{if } \theta = \text{poor} \end{cases}$$

We show the case for $\theta = \text{good}$ since the case when $\theta = \text{poor}$ is analogous. Given that $\hat{r} \in [\alpha, \beta]$, $\epsilon > 0$ and ϕ is strictly monotonically decreasing and positive in $[\alpha, \beta]$:

$$\int_{\hat{r}}^{\hat{r}+\epsilon} \phi(y)dy < \int_{\hat{r}}^{\hat{r}+\epsilon} \phi(\hat{r})dy \text{ and } \int_{\hat{r}}^{\hat{r}+\epsilon} \phi(\hat{r})dy = \int_{\hat{r}-\epsilon}^{\hat{r}} \phi(\hat{r})dy$$

and

$$\int_{\hat{r}-\epsilon}^{\hat{r}} \phi(\hat{r})dy < \int_{\hat{r}-\epsilon}^{\hat{r}} \phi(y)dy$$

Thus:

$$\int_{\hat{r}}^{\hat{r}+\epsilon} \phi(y)dy < \int_{\hat{r}-\epsilon}^{\hat{r}} \phi(y)dy \Leftrightarrow \delta(\hat{r}, \epsilon, \text{good}) > 0$$

Regarding Property 4 we need to show that the partial derivative of $\delta(\hat{r}, \epsilon, \theta)$ with respect to ϵ is greater than zero for $\theta = \text{good}$ and less than zero for $\theta = \text{poor}$, where

$$\tfrac{\partial \delta(\hat{r}, \epsilon, \theta)}{\partial \epsilon} = \begin{cases} \phi(\hat{r} - \epsilon) - \phi(\hat{r} + \epsilon), & \text{if } \theta = \text{good} \\ \\ \psi(\hat{r} - \epsilon) - \psi(\hat{r} + \epsilon), & \text{if } \theta = \text{poor} \end{cases}$$

Given that ϕ and ψ are strictly monotonically increasing and strictly monotonically decreasing, respectively, and $\hat{r} \in [\alpha, \beta]$, where $\beta > \alpha \geq 0$, and $\epsilon \in (0, \min[\hat{r} - \alpha, \beta - \hat{r}]]$: $\phi(\hat{r} + \epsilon) < \phi(\hat{r} - \epsilon)$ and $\psi(\hat{r} - \epsilon) < \psi(\hat{r} + \epsilon)$. Thus:

$$\begin{cases} \tfrac{\partial \delta(\hat{r}, \epsilon, \theta)}{\partial \epsilon} > 0, & \text{if } \theta = \text{good} \\ \\ \tfrac{\partial \delta(\hat{r}, \epsilon, \theta)}{\partial \epsilon} < 0, & \text{if } \theta = \text{poor} \end{cases}$$

Finally, Property 5 is also satisfied by (5.2) if:

$$\begin{cases} 2\phi(\hat{r}) - \phi(\hat{r} + \epsilon) - \phi(\hat{r} - \epsilon) \leq 0 \\ 2\psi(\hat{r}) - \psi(\hat{r} + \epsilon) - \psi(\hat{r} - \epsilon) \leq 0 \end{cases} \tag{5.3}$$

for $\forall \hat{r} \in [\alpha, \beta]$, where $\beta > \alpha \geq 0$, and $\forall \epsilon \in (0, \min[\hat{r} - \alpha, \beta - \hat{r}]]$. More specifically, we need to prove that the partial derivatives of $\delta(\hat{r}, \epsilon, \theta)$ with respect to $\hat{r}$ are less than or equal to zero. This is true if (5.3) holds, given that:

$$\frac{\partial\delta(\hat{r},\epsilon,\theta)}{\partial\hat{r}} = \begin{cases} 2\phi(\hat{r}) - \phi(\hat{r}+\epsilon) - \phi(\hat{r}-\epsilon) \leq 0, & \text{if } \theta = \text{good} \\ \\ 2\psi(\hat{r}) - \psi(\hat{r}+\epsilon) - \psi(\hat{r}-\epsilon) \leq 0, & \text{if } \theta = \text{poor} \end{cases}$$

Thus, if ϕ and ψ satisfy inequalities (5.3) then Property 5 is also satisfied.

Proposition 6 states that certain linear transformations of the functions ϕ and ψ can be also used to create an *i-Function.*

Proposition 6. *Let $\phi : [\alpha,\beta] \to \mathbb{R}^+$ and $\psi : [\alpha,\beta] \to \mathbb{R}^+$ be arbitrary continuous functions on $[\alpha,\beta]$, $\beta > \alpha \geq 0$. Let ϕ be strictly monotonically decreasing and let ψ by strictly monotonically increasing in $[\alpha,\beta]$, and:*

$$\begin{array}{l} 2\phi(\hat{r}) - \phi(\hat{r}+\epsilon) - \phi(\hat{r}-\epsilon) \leq 0 \\ 2\psi(\hat{r}) - \psi(\hat{r}+\epsilon) - \psi(\hat{r}-\epsilon) \leq 0 \end{array} \tag{5.4}$$

for $\forall\hat{r} \in [\alpha,\beta]$ and $\forall\epsilon \in (0, \min[\hat{r}-\alpha, \beta-\hat{r}]]$. The function:

$$I(x,\hat{r},\theta) = \begin{cases} \int_\alpha^\beta \Phi(y)dy - |\int_{\hat{r}}^x \Phi(y)dy|, & \textit{if } \theta = \text{good} \\ \\ \int_\alpha^\beta \Psi(y)dy - |\int_{\hat{r}}^x \Psi(y)dy|, & \textit{if } \theta = \text{poor} \\ \end{cases}$$

where $\Phi(y) = \lambda_1 \cdot \phi(y) + \lambda_2$, $\Psi(y) = \kappa_1 \cdot \psi(y) + \kappa_2$, $\lambda_1, \kappa_1 \in \mathbb{R}^+$, $\lambda_2, \kappa_2 \in \mathbb{R}$ and $\Phi(y), \Psi(y) \geq 0$ in $[\alpha,\beta]$, is a valid i-Function *which satisfies Properties 1 to 5.*

Proof. Given $\lambda_1, \kappa_1 \in \mathbb{R}^+$, $\lambda_2, \kappa_2 \in \mathbb{R}$ and $\phi(x)$ and $\psi(x)$ are monotonically decreasing and increasing, respectively, we have that $\Phi(x)$ is monotonically decreasing and $\Psi(x)$ is monotonically increasing. Thus, Properties 1 to 4 are satisfied. For Property 5, we just need to prove that

$$\begin{array}{l} 2\Phi(\hat{r}) - \Phi(\hat{r}+\epsilon) - \Phi(\hat{r}-\epsilon) \leq 0 \\ 2\Psi(\hat{r}) - \Psi(\hat{r}+\epsilon) - \Psi(\hat{r}-\epsilon) \leq 0 \end{array}. \tag{5.5}$$

We have that

$$\begin{aligned} 2\Phi(\hat{r}) - \Phi(\hat{r}+\epsilon) - \Phi(\hat{r}-\epsilon) &= 2(\lambda_1\phi(\hat{r}) + \lambda_2) - (\lambda_1\phi(\hat{r}+\epsilon) + \lambda_2) - (\lambda_1\phi(\hat{r}-\epsilon) + \lambda_2) \\ &= \lambda_1(2\phi(\hat{r}) - \phi(\hat{r}+\epsilon) - \phi(\hat{r}-\epsilon)) \leq 0 \end{aligned} \tag{5.6}$$

Since $2\phi(\hat{r}) - \phi(\hat{r}+\epsilon) - \phi(\hat{r}-\epsilon) \leq 0$ and $\lambda_1 \in \mathbb{R}$.

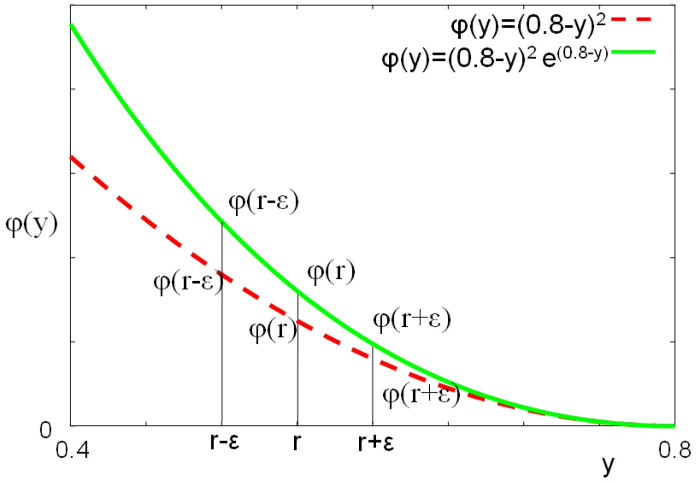


Figure 5.2: Examples of ϕ

Similarly for $\Psi(x)$ we have:

$$\begin{aligned} 2\Psi(\hat{r}) - \Psi(\hat{r}+\epsilon) - \Psi(\hat{r}-\epsilon) &= 2(\kappa_1\psi(\hat{r}) + \kappa_2) - (\kappa_1\psi(\hat{r}+\epsilon) + \kappa_2) - (\kappa_1\psi(\hat{r}-\epsilon) + \kappa_2) \\ &= \kappa_1(2\psi(\hat{r}) - \psi(\hat{r}+\epsilon) - \psi(\hat{r}-\epsilon)) \leq 0 \qquad (5.7) \end{aligned}$$

Since $2\psi(\hat{r}) - \psi(\hat{r}+\epsilon) - \psi(\hat{r}-\epsilon) \leq 0$ and $\kappa_1 \in \mathbb{R}$.

□

5.3.1 Examples of *i-Functions*

In this section we introduce two examples of an *i-Function*.

Example 1. Let $\phi(y) = (\beta - y)^n$ and $\psi(y) = y^m$. Then:

$$I(x, \hat{r}, \theta) = \begin{cases} \int_\alpha^\beta (\beta - y)^n dy - |\int_{\hat{r}}^x (\beta - y)^n dy| & \theta = \text{good} \\ \\ \int_\alpha^\beta y^m dy - |\int_{\hat{r}}^x y^m dy| & \theta = \text{poor} \end{cases} \qquad (5.8)$$

where $x, \hat{r} \in [\alpha, \beta]$, $\beta > \alpha \geq 0$, and $n, m \in \mathbb{N}^+$. An example of $\phi(y)$ with $\beta = 0.8$ and $n = 2$ is depicted in Figure 5.2.

Since ϕ and ψ are continuous, positive and strictly monotonically decreasing and strictly monotonically increasing, respectively (as can be seen by a simple check of the first and second derivatives), $I(x, \hat{r}, \theta)$ satisfies Properties 1 to Property 4. In order to prove that $I(x, \hat{r}, \theta)$ also satisfies Property 5 it is sufficient to prove that

$$2\phi(\hat{r}) - \phi(\hat{r}+\epsilon) - \phi(\hat{r}-\epsilon) = 2(\beta - \hat{r})^n - (\beta - (\hat{r}+\epsilon))^n - (\beta - (\hat{r}-\epsilon))^n \leq 0 \tag{5.9}$$

and

$$2\psi(\hat{r}) - \psi(\hat{r}+\epsilon) - \psi(\hat{r}-\epsilon) = 2\hat{r}^m - (\hat{r}+\epsilon)^m - (\hat{r}-\epsilon)^m \leq 0 \tag{5.10}$$

for $\forall \hat{r} \in [\alpha, \beta]$ and $\forall \epsilon \in (0, \min[\hat{r}-\alpha, \beta-\hat{r}]]$. Inequality (5.9) can be proved by induction. More specifically:

For $n = 1$, we have

$$2\beta - 2\hat{r} - \beta + \hat{r} - \epsilon - \beta + \hat{r} - \epsilon = -2\epsilon \leq 0$$

Assume that inequality (5.9) is true for $n = k$:

$$2(\beta - \hat{r})^k - (\beta - (\hat{r}+\epsilon))^k - (\beta - (\hat{r}-\epsilon))^k \leq 0 \tag{5.11}$$

We now prove that it is also true for $n = k+1$:

$$\begin{aligned}
& 2(\beta - \hat{r})^{k+1} - (\beta - (\hat{r}+\epsilon))^{k+1} - (\beta + (\hat{r}-\epsilon))^{k+1} \\
= \; & 2(\beta - \hat{r})^k(\beta - \hat{r}) - (\beta - (\hat{r}+\epsilon))^k(\beta - (\hat{r}+\epsilon)) - (\beta + (\hat{r}-\epsilon))^k(\beta + (\hat{r}-\epsilon)) \\
= \; & (\beta - \hat{r})(2(\beta - \hat{r})^k - (\beta - (\hat{r}+\epsilon))^k - (\beta - (\hat{r}-\epsilon))^k) + \\
+ \; & \epsilon((\beta - (\hat{r}+\epsilon))^k - (\beta - (\hat{r}-\epsilon))^k) \leq 0
\end{aligned} \tag{5.12}$$

Given inequality (5.11) the inequality (5.12) is true as it is the summation of two negative numbers. Similarly, we will prove that inequality (5.10) is also true.

For $m = 1$, we have

$$2\hat{r}^1 - (\hat{r}+\epsilon)^1 - (\hat{r}-\epsilon)^1 = 0$$

For $m = 2$, we have

$$2\hat{r}^2 - (\hat{r}+\epsilon)^2 - (\hat{r}-\epsilon)^2 = -\epsilon^2 \leq 0$$

Assume that it is true for $m = k$:

$$2\hat{r}^k - (\hat{r}+\epsilon)^k - (\hat{r}-\epsilon)^k \leq 0$$

Now, we prove that it is also true for $m = k+1$:

$$\begin{aligned} & 2\hat{r}^{k+1} - (\hat{r}+\epsilon)^{k+1} - (\hat{r}-\epsilon)^{k+1} \\ = \; & 2\hat{r}\hat{r}^{k} - (\hat{r}+\epsilon)(\hat{r}+\epsilon)^{k} - (\hat{r}-\epsilon)(\hat{r}-\epsilon)^{k} \\ = \; & \hat{r}(2\hat{r}^{k} - (\hat{r}+\epsilon)^{k} - (\hat{r}-\epsilon)^{k}) + \epsilon((\hat{r}-\epsilon)^{k} - (\hat{r}+\epsilon)^{k}) \leq 0 \qquad (5.13) \end{aligned}$$

Inequality (5.13) is true since it is the summation of two negative numbers.

Example 2. In the second example we consider $\phi(y) = (\beta - y)^n e^{(\beta - y)}$ and $\psi(y) = y^m e^y$. In this case the *i-Function* (5.2) will become:

$$I(x, \hat{r}, \hat{\theta}) = \begin{cases} \int_{\alpha}^{\beta} (\beta - y)^n e^{(\beta-y)} dy - |\int_{\hat{r}}^{x} (\beta - y)^n e^{(\beta-y)} dy| & \theta = \text{good} \\ \\ \int_{\alpha}^{\beta} y^m e^y dy - |\int_{\hat{r}}^{x} y^m e^y dy| & \theta = \text{poor} \end{cases} \qquad (5.14)$$

An example of ϕ for $\beta = 0.8$ and $n = 2$ is depicted in Figure 5.2.

Similarly with the first example, since ϕ and ψ are continuous, positive, and strictly monotonically decreasing and strictly monotonically increasing, respectively (as can be seen by a simple check of the first and second derivatives), $I(x, \hat{r}, \theta)$ satisfies Properties 1 to 4.

As in the case of the first family, in order to prove that *i-Function* (5.14) also satisfies Property 5 it is sufficient to show that $\forall n, m \in \mathbb{N}$, $\forall \hat{r} \in [\alpha, \beta]$ and $\forall \epsilon \in (0, \min[\hat{r}-\alpha, \beta-\hat{r}]]$:

$$\begin{aligned} 2\phi(\hat{r}) - \phi(\hat{r}+\epsilon) - \phi(\hat{r}-\epsilon) = \\ = 2(\beta - \hat{r})^n e^{\beta - \hat{r}} - (\beta - (\hat{r}+\epsilon))^n e^{(\beta - (\hat{r}+\epsilon))} - (\beta - (\hat{r}-\epsilon))^n e^{(\beta - (\hat{r}-\epsilon))} \leq 0 \\ \Leftrightarrow 2(\beta - \hat{r})^n - (\beta - (\hat{r}+\epsilon))^n e^{-\epsilon} - (\beta - (\hat{r}-\epsilon))^n e^{\epsilon} \leq 0 \qquad (5.15) \end{aligned}$$

and

$$\begin{aligned} 2\psi(\hat{r}) - \psi(\hat{r}+\epsilon) - \psi(\hat{r}-\epsilon) \;&=\; 2\hat{r}^m e^{\hat{r}} - (\hat{r}+\epsilon)^m e^{\hat{r}+\epsilon} - (\hat{r}-\epsilon)^m e^{\hat{r}-\epsilon} \leq 0 \\ &\Leftrightarrow\; 2\hat{r}^m - (\hat{r}+\epsilon)^m e^{\epsilon} - (\hat{r}-\epsilon)^m e^{-\epsilon} \leq 0 \qquad (5.16) \end{aligned}$$

Consider the case where $\theta =$ poor. We will prove that inequality (5.16) is true by induction.[2] For $m = 0$ we have: $2 - e^{\epsilon} - e^{-\epsilon}$ which is clearly less or equal to zero. For $m = 1$ we have:

[2] Given that $e^r > 0$ we can omit it.

$$2\hat{r} - e^{\epsilon}(\hat{r}+\epsilon) - e^{-\epsilon}(\hat{r}-\epsilon) = \epsilon(e^{-\epsilon} - e^{\epsilon}) + (2 - e^{\epsilon} - e^{-\epsilon}) \leq 0$$

which is true since it is the summation of two negative numbers. Assume now that (5.16) is true for $m = k$:

$$2\hat{r}^k - e^{\epsilon}(\hat{r}+\epsilon)^k - e^{-\epsilon}(\hat{r}-\epsilon)^k \leq 0 \tag{5.17}$$

We will prove that it is also true for $m = k+1$.

$$\begin{aligned} 2\hat{r}^{k+1} - e^{\epsilon}(\hat{r}+\epsilon)^{k+1} - e^{-\epsilon}(\hat{r}-\epsilon)^{k+1} \leq 0 \Leftrightarrow \\ -e^{\epsilon}(\hat{r}+\epsilon)^k(\hat{r}+\epsilon) + 2\hat{r}^k\hat{r} - e^{-\epsilon}(\hat{r}-\epsilon)^k(\hat{r}-\epsilon) \leq 0 \Leftrightarrow \\ (-e^{\epsilon}(\hat{r}+\epsilon)^k + 2\hat{r}^k - e^{-\epsilon}(\hat{r}-\epsilon)^k)\hat{r} + \epsilon(e^{-\epsilon}(\hat{r}-\epsilon) - e^{\epsilon}(\hat{r}+\epsilon)) \leq 0 \end{aligned}$$

which is true since it is the summation of two negative numbers. Consequently, the inequality (5.16) holds for $\forall m \in \mathbb{N}$.

Consider the case where $\theta =$ good. We will prove that inequality (5.15) is true by induction.[3]

For $n = 1$ we have:

$$2(\beta - \hat{r}) - (\beta - (\hat{r}+\epsilon))e^{-\epsilon} - (\beta - (\hat{r}-\epsilon))e^{\epsilon} = (\beta - \hat{r})(2 - e^{\epsilon} - e^{-\epsilon}) \leq 0$$

which is clearly less or equal to zero since $(\beta - \hat{r}) \geq 0$ and $(2 - e^{\epsilon} - e^{-\epsilon}) \leq 0$.

Assume now that (5.16) is true for $n = k$:

$$2(\beta - \hat{r})^k - (\beta - (\hat{r}+\epsilon))^k e^{-\epsilon} - (\beta - (\hat{r}-\epsilon))^k e^{\epsilon} \leq 0$$

We will prove that it is also true for $n = k+1$.

$$\begin{aligned} 2(\beta - \hat{r})^{k+1} - (\beta - (\hat{r}+\epsilon))^{k+1} e^{-\epsilon} - (\beta - (\hat{r}-\epsilon))^{k+1} e^{\epsilon} \leq 0 \Leftrightarrow \\ (\beta - \hat{r})(2(\beta - \hat{r})^k - (\beta - (\hat{r}+\epsilon))^k e^{-\epsilon} - (\beta - (\hat{r}-\epsilon))^k e^{\epsilon}) + \epsilon((\beta - \hat{r} - \epsilon)e^{-\epsilon} - (\beta - \hat{r} + \epsilon)e^{\epsilon}) \leq 0 \end{aligned} \tag{5.18}$$

which is true since it is the summation of two negative numbers. Consequently, the inequality (5.15) holds for $\forall n \in \mathbb{N}$.

[3] Given that $e^r > 0$ we can omit it.

Choosing ϕ and ψ

In order to find the most suitable instances of the above families of functions a number of different criteria could be used. For example, one criterion is bounding the maximum value of $\delta(\hat{r}, \epsilon, \theta)$.

More specifically, consider the case of $\theta =$ good, a criterion could be to choose the function ϕ in such a way that the maximum value of the $\delta(\hat{r}, \epsilon, \theta)$ (which essentially defines the maximum difference of the *i-Scores* of two ratings that deviate equally from the rating $\hat{r}$ only by absolute value) is bounded by U and L, where $U, L \in \mathbb{R}^+$, in order to avoid over-penalizing communities. A simple way of finding ϕ and ψ that satisfy the following bounds:

$$L \leq max[\delta(\hat{r}, \epsilon, good)] \leq U \text{ and } L' \leq min[\delta(\hat{r}, \epsilon, poor)] \leq U'$$

in an interval $[w_1, w_2]$, where $U, L, w_1, w_2 \in \mathbb{R}^+$ and $U', L' \in \mathbb{R}^-$ is presented in Proposition 7.

Proposition 7. *Given an upper bound U and a lower bound L finding the function ϕ such that for $\forall \epsilon \in (0, min[w_2 - \hat{r}, \hat{r} - w_1]]$ and $\forall \hat{r} \in [w_1, w_2]$:*

$$L \leq max[\delta(\hat{r}, \epsilon, good)] \leq U$$

is equivalent to finding the function ϕ such that:

$$L \leq \int_{\hat{r}-\overline{\epsilon}}^{x} \phi(y)dy \leq U$$

where $\overline{\epsilon} = min[w_2 - \hat{r}, \hat{r} - w_1]$, and x is the solution to: $I(\hat{r} + \overline{\epsilon}, \hat{r}, good) = I(x, \hat{r}, good)$. While finding the function ψ that for $\forall \epsilon \in (0, min[w_2 - \hat{r}, \hat{r} - w_1]]$ and $\forall \hat{r} \in [w_1, w_2]$

$$L' \leq min[\delta(\hat{r}, \epsilon, poor)] \leq U'$$

is equivalent to finding the function ψ such that:

$$L' \leq \int_{x}^{\hat{r}+\overline{\epsilon}} \psi(y)dy \leq U'$$

where $\overline{\epsilon} = min[w_2 - \hat{r}, \hat{r} - w_1]$, and x is the solution to: $I(\hat{r} - \overline{\epsilon}, \hat{r}, poor) = I(x, \hat{r}, poor)$.

Proof. Consider the case where $\theta = good$.[4] Given that $\delta(\hat{r}, \epsilon, good)$ is monotonically increasing in ϵ: i) it is maximized when $\epsilon = min[\hat{r} - w_1, w_2 - \hat{r}]$, and ii) there will be a $x \in (\hat{r} - \epsilon, \hat{r})$ such that $I(x, \hat{r}, good) = I(\hat{r} + \overline{\epsilon}, \hat{r}, good)$. Thus, we can write $\delta(\hat{r}, \epsilon, \theta)$ as:

$$\delta(\hat{r}, \epsilon, \theta) = \int_{\hat{r}-\overline{\epsilon}}^{x} \phi(y)dy$$

□

Essentially, Proposition 7 states that in order to find a ϕ that leads to an $\delta(\hat{r}, \epsilon, \theta)$ whose maximum value is between two bounds L and U, we just have to find a ϕ such that:

$$L \leq \int_{\hat{r}-\overline{\epsilon}}^{x} \phi(y)dy \leq U$$

where $\overline{\epsilon} = min[w_2 - \hat{r}, \hat{r} - w_1]$, and x is the solution to: $I(\hat{r} + \overline{\epsilon}, \hat{r}, good) = I(x, \hat{r}, good)$. Analogous is the case for the function ψ.

5.3.2 Example of *i-Score*-based Payment Function

Let (5.8) be our *i-Function*. For $\theta = good$ and $[\alpha, \beta] = [0, 1]$ we have

$$I(x, \hat{r}, \theta) = \begin{cases} \frac{1}{n+1} - \frac{(1-x)^{n+1}}{n+1} + \frac{(1-\hat{r})^{n+1}}{n+1} & x \geq \hat{r} \\ \frac{1}{n+1} - \frac{(1-\hat{r})^{n+1}}{n+1} + \frac{(1-x)^{n+1}}{n+1} & x < \hat{r} \end{cases} \tag{5.19}$$

For $\theta = poor$ and $[\alpha, \beta] = [0, 1]$ we have

$$I(x, \hat{r}, \theta) = \begin{cases} \frac{1}{n+1} - \frac{x^{n+1}}{n+1} + \frac{\hat{r}^{n+1}}{n+1} & x \geq \hat{r} \\ \frac{1}{n+1} + \frac{x^{n+1}}{n+1} - \frac{\hat{r}^{n+1}}{n+1} & x < \hat{r} \end{cases} \tag{5.20}$$

Graphical representations of the *i-Functions* in (5.19) and (5.20) for $n = 2$ are presented in Figures 5.3 & 5.4, respectively.

Consider now that when the communities provide information they have to share an amount κ based on their *i-Score*. In particular, we define the payment $P(r_j^k, \hat{r}, \theta_j^k)$ that each community $C_k \in m(a_j)$ will receive as

[4] The proof for θ = poor is analogous.

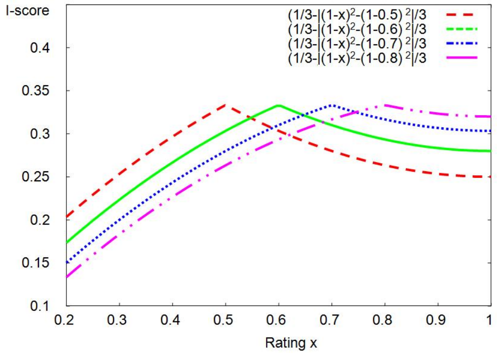


Figure 5.3: *i-Function* example for n=2, $\hat{r} = \{0.5, 0.6, 0.7, 0.8, 0.9\}$ and type *good*

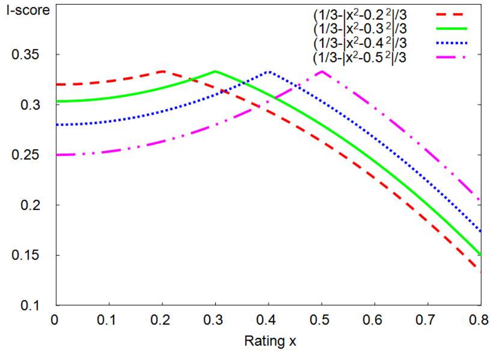


Figure 5.4: *i-Function* example for n=2, $\hat{r} = \{0.2, 0.3, 0.4, 0.5\}$ and type *poor*

$$P(r_j^k, \hat{r}, \theta_j^k) = I(r_j^k, \hat{r}, \theta_j^k))^d \cdot \gamma \tag{5.21}$$

where

$$\gamma = \frac{\kappa}{\sum_{\forall C_k \in m(a_j)} (I(r_j^k, \hat{r}, \theta_j^k))^d}$$

Thus, the payment function is

$$P(I(r_j^k, \hat{r}, good)) = (I(r_j^k, \hat{r}, \theta_j^k))^d \cdot \frac{\kappa}{\sum_{\forall C_k \in m(a_j)} (I(r_j^k, \hat{r}, \theta_j^k))^d} \tag{5.22}$$

where $d \in \mathbb{Z}$.

Given that the *i-Function* is positive the function $P(I(r_j^k, \hat{r}, good))$ is monotonically increasing with respect to the *i-Score* $I(r_j^k, \hat{r}, good)$. For example, if a community C_k receives a higher score than community C_j, it will also receive a higher payment. Essentially, the *i-Function* determines the order of the *i-Scores* and the distance between each pair of *i-Scores*. By choosing $d = 1$ we use directly the *i-Scores* for determining the payments.

As we will see in the following examples the higher the d the larger the difference in the payment between honest agents and agents that are less honest. However, we can also control the variance of the payments by choosing an appropriate *i-Function*.[5]

In the following sections we provide an example that demonstrates the *i-Scores* and the payment the advisor communities will receive by using our suggested *i-Functions*.

Example.

Assume now that community C_i requests information from the communities $m(a_2) = \{C_a, C_b, C_c, C_d, C_e, C_f, C_g, C_h\}$ about an agent a_2. The communities in $m(a_2)$ provided the following information:

- C_a: $(r_2^a = 0.2, \theta_2^a = \mathit{poor})$,
- C_b: $(r_2^b = 0.3, \theta_2^b = \mathit{poor})$,
- C_c: $(r_2^c = 0.4, \theta_2^c = \mathit{poor})$,
- C_d: $(r_2^d = 0.5, \theta_2^d = \mathit{poor})$,
- C_e: $(r_2^e = 0.5, \theta_2^e = \mathit{good})$,
- C_f: $(r_2^f = 0.6, \theta_2^f = \mathit{good})$,
- C_g: $(r_2^g = 0.7, \theta_2^g = \mathit{good})$,
- C_h: $(r_2^h = 0.8, \theta_2^h = \mathit{good})$.

Poor Contributor Scenario

Assume that the community C_i decided to accept the agent a_2 which proved to have a type $\theta = \mathit{poor}$.[6] Table 5.1 presents the *i-Score* each community will receive for any of the ratings $\hat{r} \in \{0.2, 0.3, 0.4, 0.5\}$ the community C_i might experience. For example, if the

[5] Recall, we consider that the *honest* communities can accurately reason about the behaviour of their agents and that their agents do not change their behaviour from one community to another. In addition we consider that a community which does not (strategically or not) or cannot reason about the value of its agents is *dishonest*.

[6] Note, the mechanism for determining whether to accept or not the agent is beyond the scope of this thesis.

$\hat{r}$	$C_a(.2, pr)$	$C_b(.3, pr)$	$C_c(.4, pr)$	$C_d(.5, pr)$	$C_i\ \forall i \in \{e, f, g, h\}$
0.5	0.294333	0.300667	0.313000	0.333333	0
0.4	0.314667	0.321000	0.333333	0.313000	0
0.3	0.327000	0.333333	0.321000	0.300667	0
0.2	0.333333	0.327000	0.314667	0.294333	0

Table 5.1: Example-Poor Contributor: i-Scores for $n = 2$

$\hat{r}$	$C_a(.2, pr)$	$C_b(.3, pr)$	$C_c(.4, pr)$	$C_d(.5, pr)$	$C_i\ \forall i \in \{e, f, g, h\}$
0.5	1.9	1.94	2.02	2.15	0
0.4	1.97	2.00	2.08	1.95	0
0.3	2.04	2.08	2.00	1.88	0
0.2	2.10	2.06	1.98	1.86	0

Table 5.2: Example-Poor Contributor: Payment Distribution for $d = 1$ and $n = 2$

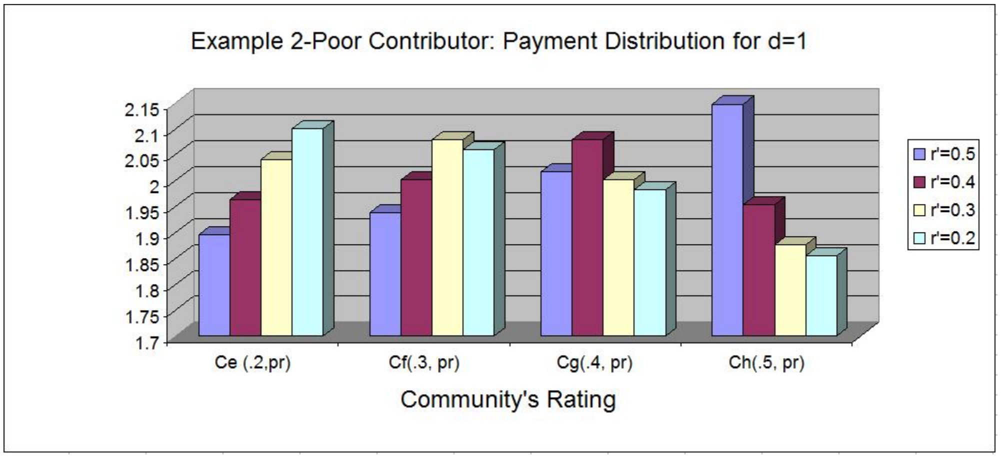


Figure 5.5: Example-Poor Contributor: Payment Distribution for $d = 1$ and $n = 2$

community C_i experiences a rating $\hat{r} = 0.3$, the community C_b that provided a rating equal to 0.3 will receive the maximum possible score which is 0.333333, while the communities C_a and C_c that provided ratings 0.2 and 0.4 will receive 0.313 and 0.321, respectively. We assign an *i-Score* equal to 0 to all the communities which misclassified the type of the agent. In this case, this includes the communities that assigned a type equal to *good* (i.e., $\{C_e, C_f, C_g, C_h\}$).

Let \$8 be the amount the community C_i has allocated to pay the communities in $m(a_2)$. We have $\kappa = 8$. If all the communities in $m(a_2)$ provide accurate information they will receive \$1 each. However, since the communities C_d, C_e, C_f, and C_g influenced the community C_i towards making the wrong decision and accept an agent that is a poor contributor they will not receive any payment. Thus, the amount κ will be distributed

$\hat{r}$	$C_a(.2, pr)$	$C_b(.3, pr)$	$C_c(.4, pr)$	$C_d(.5, pr)$	$C_i \ \forall i \in \{e, f, g, h\}$
0.5	1.50	1.67	2.04	2.79	0
0.4	1.81	2.00	2.42	1.77	0
0.3	2.18	2.40	1.99	1.43	0
0.2	2.50	2.28	1.88	1.34	0

Table 5.3: Example-Poor Contributor: Payment Distribution for $d = 5$ and $n = 2$

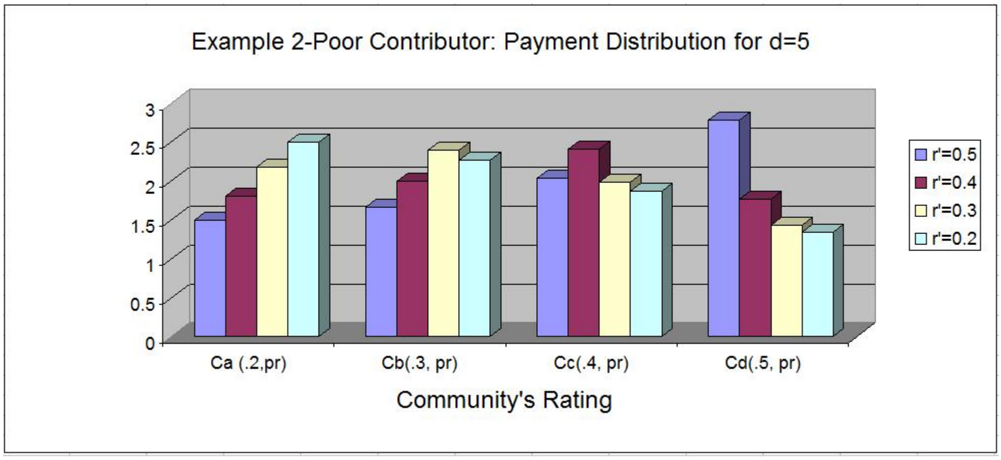


Figure 5.6: Example-Poor Contributor: Payment Distribution for $d = 5$ and $n = 2$

$\hat{r}$	$C_a(.2, pr)$	$C_b(.3, pr)$	$C_c(.4, pr)$	$C_d(.5, pr)$	$C_i \ \forall i \in \{e, f, g, h\}$
0.5	1.06	1.31	1.96	3.67	0
0.4	1.62	1.97	2.88	1.53	0
0.3	2.30	2.79	1.91	1.00	0
0.2	2.99	2.47	1.68	0.86	0

Table 5.4: Example-Poor Contributor: Payment Distribution for $d = 10$ and $n = 2$

among the communities C_a, C_b, C_c and C_d based on their *i-Score* (Table 5.1).

Tables 5.2, 5.3 and 5.4 depict the distribution of the amount κ using the payment function (5.22) for $d = 1, 5, 10$ and $n = 2$. Figures 5.5, 5.6 and 5.7 depict the graphical representation of the payments distribution for $d = 1$, $d = 5$ and $d = 10$, respectively.[7]

As we can see, the community that correctly classified the agent as poor and reported a rating $\hat{r}$ equal to the one C_i experiences will receive the highest payment for any $\hat{r}$. In addition, the communities that deviated by an amount $|\epsilon|$ will receive higher payment if their rating is equal to $\hat{r} - |\epsilon|$ from the communities whose rating is $\hat{r} + |\epsilon|$. Furthermore, similarly to *Example 1*, as d increases the variance of the distribution of the amount κ

[7] Note that $r' = \hat{r}$.

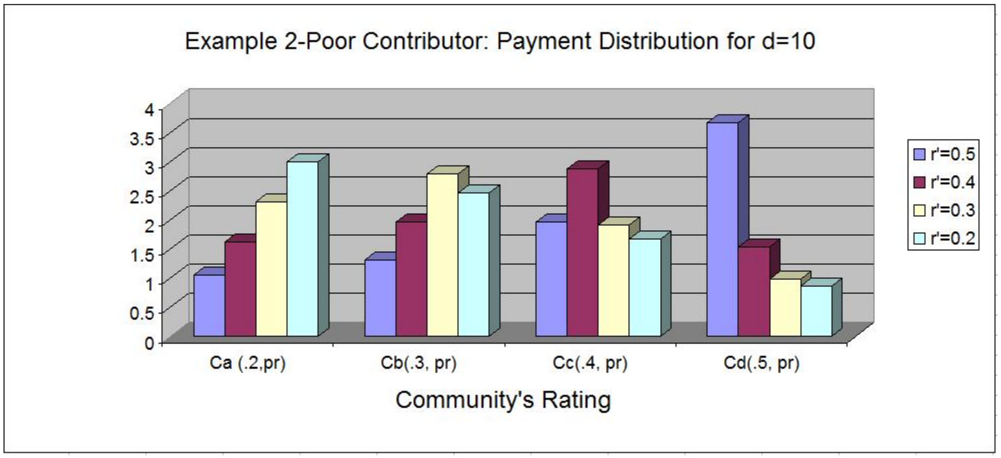


Figure 5.7: Example-Poor Contributor: Payment Distribution for $d = 10$ and $n = 2$

$\hat{r}$	$C_e(.5, gd)$	$C_f(.6, gd)$	$C_g(.7, gd)$	$C_h(.8, gd)$	$C_i\ \forall i \in \{a, b, c, d\}$
0.8	0.294333	0.314667	0.327000	0.333333	0
0.7	0.300667	0.321000	0.333333	0.327000	0
0.6	0.313000	0.333333	0.321000	0.314667	0
0.5	0.333333	0.313000	0.300667	0.294333	0

Table 5.5: Example-Good Contributor: i-Scores for $n = 2$

increases. For example, if $\hat{r} = 0.4$ then the community C_c will receive \$2.08 for $d = 1$, \$2.42 for $d = 5$, and \$2.88 for $d = 10$. The community C_b that reported a rating 0.3 will receive \$2.00 for $d = 1$, \$2.00 for $d = 5$, and \$1.97 for $d = 10$. While the community C_d that reported a rating 0.5 will receive \$1.95 for $d = 1$, \$1.77 for $d = 5$, and \$1.53 for $d = 10$.

$\hat{r}$	$C_e(.5, gd)$	$C_f(.6, gd)$	$C_g(.7, gd)$	$C_h(.8, gd)$	$C_i\ \forall i \in \{a, b, c, d\}$
0.8	1.86	1.98	2.06	2.10	0
0.7	1.88	2.00	2.08	2.04	0
0.6	1.95	2.08	2.00	1.97	0
0.5	2.15	2.02	1.94	1.90	0

Table 5.6: Example-Good Contributor: Payment Distribution for $d = 1$ and $n = 2$

$\hat{r}$	$C_e(.5, gd)$	$C_f(.6, gd)$	$C_g(.7, gd)$	$C_h(.8, gd)$	$C_i\ \forall i \in \{a, b, c, d\}$
0.8	1.34	1.88	2.28	2.50	0
0.7	1.43	1.99	2.40	2.18	0
0.6	1.77	2.42	2.00	1.81	0
0.5	2.79	2.04	1.67	1.50	0

Table 5.7: Example-Good Contributor: Payment Distribution for $d = 5$ and $n = 2$

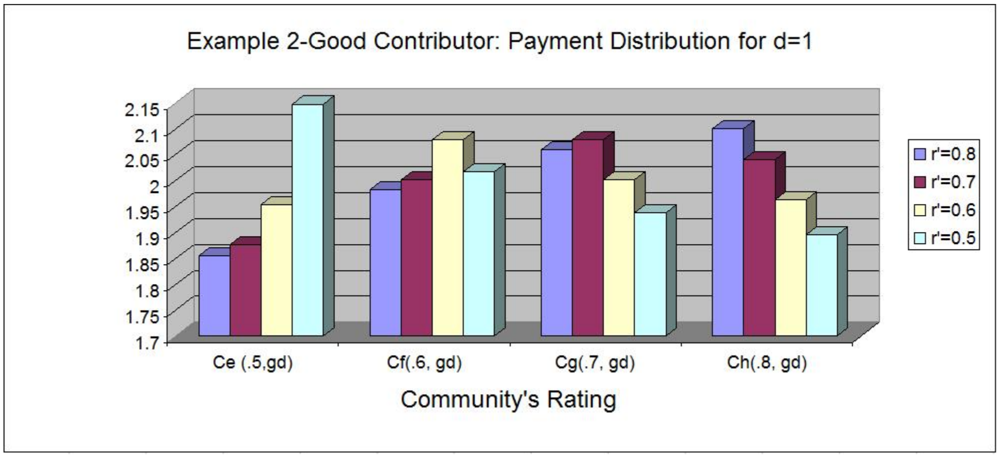


Figure 5.8: Example-Good Contributor: Payment Distribution for $d = 1$ and $n = 2$

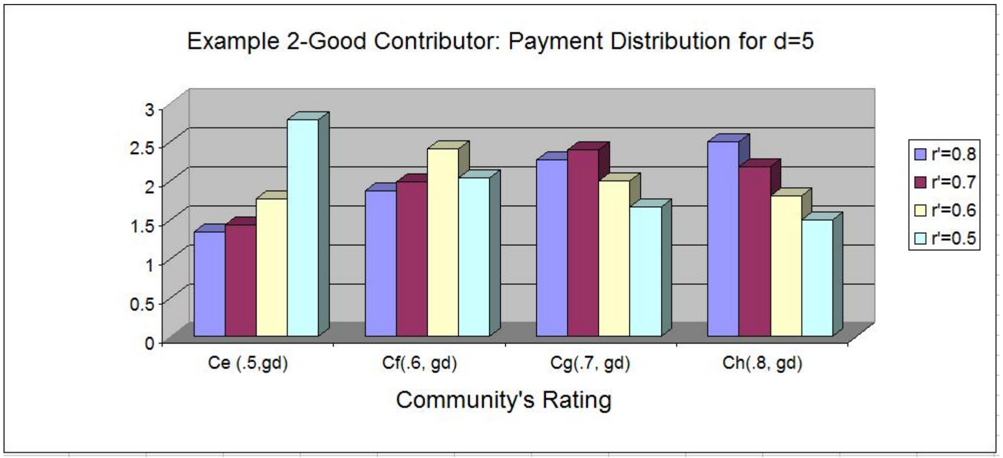


Figure 5.9: Example-Good Contributor: Payment Distribution for $d = 5$ and $n = 2$

$\hat{r}$	$C_e(.5, gd)$	$C_f(.6, gd)$	$C_g(.7, gd)$	$C_h(.8, gd)$	$C_i\ \forall i \in \{a, b, c, d\}$
0.8	0.86	1.68	2.47	2.99	0
0.7	1.00	1.91	2.79	2.30	0
0.6	1.53	2.88	1.97	1.62	0
0.5	3.67	1.96	1.31	1.06	0

Table 5.8: Example-Good Contributor: Payment Distribution for $d = 10$ and $n = 2$

Good Contributor Scenario

Assume now that the community C_i decides to accept the agent a_2 and a_2 proves to have a type $\theta = good$. Table 5.5 presents the *i-Score* each community will receive for

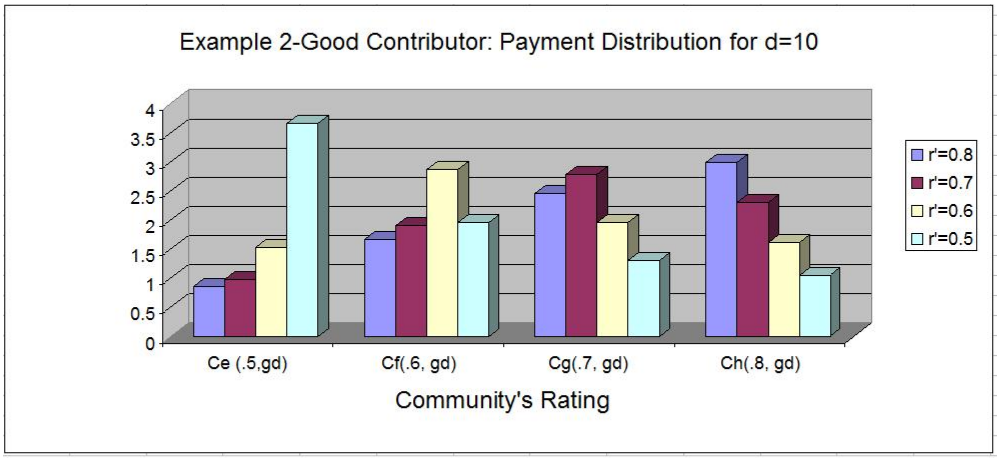


Figure 5.10: Example-Good Contributor: Payment Distribution for $d = 10$ and $n = 2$

any of the following ratings $\hat{r} \in \{0.5, 0.6, 0.7, 0.8\}$ the community C_i might experience. Recall, we assign an *i-Score* equal to 0 to all the communities which misclassified the type of the agent. In this case, this is the communities that assigned a type equal to *poor* (i.e., $\{C_a, C_b, C_c, C_d\}$). Also, these communities are not going to get paid since they influenced the community C_i to make the wrong decision and accept an agent that is a poor contributor. Thus, the amount κ will be distributed among the communities C_e, C_f, C_g and C_h based on their *i-Score*.

Tables 5.6, 5.7 and 5.8 depict the distribution of the amount κ using the payment function (5.22) for $n = 2$ and $d = 1$, $d = 5$ and $d = 10$, respectively Figures 5.9 & 5.10 depict the graphical representation of the payments distribution for $d = 1$, $d = 5$ and $d = 10$, respectively.

For example, if $\hat{r} = 0.7$ then the community C_g will receive \$2.08 for $d = 1$, \$2.4 for $d = 5$, and \$2.79 for $d = 10$. The community C_f that reported a rating 0.6 will receive \$2.00 for $d = 1$, \$2.00 for $d = 5$, and \$1.91 for $d = 10$. While the community C_h that reported a rating 0.8 will receive \$2.08 for $d = 1$, \$2.18 for $d = 5$, and \$2.3 for $d = 10$.

5.4 Summary

Summarizing, in this chapter we presented the properties a scoring function, that determines compensation a party will receive when it provides evaluation information, should follow. The compensation takes into consideration both the ratings and their interpretation. We set our scoring function to maximize the payment of a community only when it provides a truthful rating and we introduced a set of properties a scoring function should follow in order to promote honesty and fairness. The key idea of the properties is as follows: If the reported rating deviates from the rating $\hat{r}$ (the recipient community experiences) and influences the recipient community towards making the wrong decision (i.e., decline a good contributor or accept a poor contributor), the advisor community that provided this rating should receive a lower payment than if it had reported a rating of equal deviation but which was influencing towards the right decision (i.e., accept a good contributor or decline a poor contributor). Furthermore, we showed that such scoring function exists and we provided examples of specific families of functions that can be used. Finally, we presented directions of possible criteria in selecting a specific instance of the latter families.

Chapter 6

The Trust Modeling Problem

As Binmore & Dasgupta stated [6] we have significantly limited knowledge on how people acquire trust. Thus, we believe that trying to provide a general model of trust for a multiagent system in which agents are controlled by and act on behalf of users is very ambitious. For this reason, in our work we focus on modeling trust in terms of promises. In particular, in this chapter we present the minimum requirements of a framework for reasoning about the trustworthiness of agents who contribute resources in settings where important decisions are made based on these resources and thus their truthful disclosure is crucial. Examples of these settings include planning [7], task allocation [56] etc. Our ultimate goal is to discourage the agents from both over-stating and under-stating their potential contributions.

Consider the following motivating examples. A set of underwater robots belong to different companies and collaborate to complete tasks like repairing cables and pipes or stopping the leak of sinked ships' oil tanks. Each robot is represented by an agent and has to disclose its capabilities in order to generate a plan that will allow all the robots to collaborate efficiently and effectively for completing their mission. Since the cost of the operation for this type of robot is high, the agents have incentives to exhibit strategic behaviour. They might try to over-declare the capabilities of the robots they represent to ensure their participation in the mission or they might under-declare their capabilities hoping that another robot will do the costly or risky tasks. These agents only reveal the true capabilities of the robots they represent, when the time for the contribution of the robot arrives, or when the plan fails or if the payoff of the plan turns out to be lower than their expectation. However, if agents truthfully reveal what they are capable of delivering, then the best plan can be found and the robots can be deployed in the most effective manner. We thus want a trust model which can be used to judge the reliability of the agents and also to encourage them to be reliable.

Another example is the following: Consider a set of agents A that are interested in selling a set of tickets through an online company. The company offers a number of different plans (i.e., reduced posting fee, access to reduced shipping fee) depending on the number of tickets the sellers offer. The agents in A can form groups in order to achieve their participation in a better plan. However, some agents can behave strategically and lie regarding the number of tickets they can provide hoping that other agents will provide a sufficient number of tickets while they will be able push their extra tickets through a better deal. If the agents have the flexibility to increase their contribution after a plan has been selected, the deceptive agents can push their tickets if their other deals failed pretending that they were honestly not aware of these extra tickets.

In our model each agent promises to contribute a certain amount of resources/services and its trustworthiness depicts the extent that it deviates from its promise. Since we target settings where the exact knowledge of the resources a member can contribute is crucial for collective decisions we aim to discourage agents from strategic behaviour (i.e., over-stating and under-stating their resources). Our proposed model is incentive compatible in that agents maximize their trust scores by delivering what they promise. The agents are interested in maximizing their trustworthiness because this would maximize their chances of participating in beneficial collaborations (i.e., be chosen as the as the seller to interact with in systems like ebay.com). However, we argue that this alone is not sufficient. A promise-based trust model should be able to cope with different deviations from the initial promise. Furthermore, it should guarantee that an agent cannot benefit at the expense of others, by failing to deliver what it promised. As we will show, our model provides this guarantee. Here, we would like to note that we chose to develop a trust model that results in continuous instead of discete ratings, since we argue that this provides a more accurate and flexible tool to depict the trustworthiness of an agent. Furthermore, a continuous rating can be easily converted to a discrete rating without loss of information. This can be achieved by defining the intervals that correspond to each discrete value the rating can receive.

For example, consinder a discrete evaluation model in which the domain of the input and output is the set $\{Bad, OK, Good, VeryGood, Excellent\}$. Assume we would like to use this discrete model to determine the overal rating of two agents a_1 and a_2. Assume that the agent a_1 received four ratings, 2 *Excellent* and 2 *Bad* and the agent a_2 received 4 ratings *Good*. In the case of the agent a_2 it is straight forward that the output rating should be *Good*, however this is not the case for the agent a_1. In particular, it is not clear what should be the overall rating for a_1. More specifically, we argue that none of the ratings in $\{Bad, OK, Good, VeryGood, Excellent\}$ is guarranteed that it can depict accurately the value of the agent a_1, since the real value of the agent can be something in between.

The rest of the chapter is organized as follows. We first introduce our model, we describe the properties our trust function should exhibit and we present an instantiation. We show that these properties ensure that an agent maximizes its trust-score by both delivering what it promised and by promising what it can deliver. We also discuss how our model can be adapted to settings where, due to changing circumstances, an agent may be unsure when they make a promise that they will be able to deliver as desired. In particular, we define conditions under which an agent can update its promise without harming its trustworthiness.

6.1 The Model

Let C denote a community and let a represent an agent which is interested in joining the community C. There are many reasons why the agent a may be interested in joining C, including wishing to be able to partake in resources and opportunities available to community members, or to belong to a group of like-minded individuals. In exchange for becoming a member of the community, C requests that a contributes to the community by offering *services*, S. For example, a community may request that an agent provides CPU, bandwidth, disk space or even information. Agent a selects a service, $s \in S$ and then makes a *promise* to the community that it will contribute a particular level of that service.

Let $pm(s, a)$ be the promised level of service that agent a makes to the community. We normalize $pm(s, a)$ so that $pm(s, a) \in [0, 1]$. Up to some agreed-upon time period, T, the agent is required to *deliver* its contribution. We denote the actual delivered contribution of service s by agent a by $dl(s, a) \in [0, 1]$. Once a community has observed the delivered contribution of the agent, it can compare them to the promised contribution. This comparison results in the community assigning the agent a *trust* score, $Tr(pm(s, a), dl(s, a)) \in \mathbb{R}^+$ which depends on whether the agent delivered what it promised.

6.2 Promise-Based Trust

In this section we describe our trust model which is based on a comparison between the promises made, and the level of services delivered by a particular agent, a.

Given our basic framework, there are several key properties which we believe the trust function should exhibit:

1. An agent that delivers what it promises should be considered to be trustworthy.
2. An agent that delivers *more* than it promised should be considered to be more trustworthy than an agent that delivers *less* than it promised, assuming all other things are equal.

3. If two agents both deliver less than they promised, then the one that delivered closest to its promise should be considered to be more trustworthy. Similarly, if two agents both deliver more than they promised, then the agent that delivers closest to its promise should be considered to be more trustworthy.

Our fourth property below is an extension of our third desired property and allows us to reason about what to do if two agents both under-contribute or over-contribute equally. We explicitly state it since we will find its formalization useful later in the chapter.

4. Assume two agents under-delivered on their promise by the same amount. Then the one who promised *less* should be considered to be more trustworthy. Similarly if two agents both over-delivered on their promise by the same amount, the one who promised *more* should be considered to be more trustworthy.

To capture the four properties, we propose a trust function, Tr, which is *decomposable* into two separate components. The first component measures whether the agent has actually delivered what it promised, and we capture this by using a *reliability function*, RL. The second component measures the quality of the delivered contribution of the agent to the group, and we capture this be using a *quality-of-contribution function*, QoC. We initially do not specify how the trust function is composed of the reliability and quality-of-contribution function and instead state merely that

$$Tr(pm(s,a), dl(s,a)) = RL(pm(s,a), dl(s,a)) \diamond QoC(pm(s,a), dl(s,a))$$

where $\diamond$ is some unspecified operation. We provide a particular instantiation later in this chapter.

6.2.1 The Reliability Function

The reliability function $RL : [0,1] \times [0,1] \mapsto \mathbb{R}^+$ measures whether an agent, a, has delivered the service, s, that it promised. In particular, we say that an agent is reliable if $pm(s,a) = dl(s,a)$ and aim to discourage the agent from either under- or over-delivering on their promise. Thus, we argue that reasonable candidates for the reliability function, $RL(pm(s,a), dl(s,a))$, are functions which are maximized when $pm(s,a) = dl(s,a)$. We place the additional condition that for a promise $pm(s,a)$, the function should be strictly monotonically increasing for $0 \leq dl(s,a) \leq pm(s,a)$ and strictly monotonically decreasing for $pm(s,a) \leq dl(s,a) \leq 1$.

6.2.2 The Quality-of-Contribution Function

The quality-of-contribution function, $QoC : [0, 1] \times [0, 1] \mapsto \mathbb{R}^+$, measures the quality of the delivered service. There are two key properties we believe candidates for the QoC function should exhibit. First, we believe that larger contributions should be considered better than smaller contributions given a promise. We can formalize this property as follows;

QoC Property 1. *For any promise $pm(s, a)$, $QoC(pm(s, a), 0) = 0$, and for $dl(s, a) > 0$, $QoC(pm(s, a), dl(s, a))$ is positive and strictly monotonically increasing in $dl(s, a)$.*

The second property we endorse is the following. If two agents both *over-deliver* on their respective promises by some amount ϵ, then the one who initially promised more will be deemed to be making a better contribution. Similarly, if two agents both *under-deliver* on their respective promises by some amount ϵ then the one who initially promised less is deemed to be making a better contribution, since the expectations about the agent were less to start with. Formally

QoC Property 2. *Assume that agent a made promise $pm(s, a)$. Then for any $\epsilon > 0$ and for any $p' < pm(s, a)$*

$$QoC(pm(s, a), pm(s, a) + \epsilon) > QoC(p', p' + \epsilon)$$

and

$$QoC(pm(s, a), pm(s, a) - \epsilon) < QoC(p', p' - \epsilon).$$

6.2.3 The Trust Function

At the start of this section we outlined four properties we believe a promise-based trust function should have. In this subsection we formalize these properties for a function $Tr : [0, 1] \times [0, 1] \mapsto \mathbb{R}^+$. First, we argued that an agent who delivers what it promises is trustworthy. We place an additional constraint on the function so as to ensure that the trust function is maximized only when it delivers what it promised.

Trust Property 1. *For any promise $pm(s, a)$, Tr is maximized at $Tr(pm(s, a), pm(s, a))$. Additionally, for a given promise $pm(s, a)$, $Tr(pm(s, a), dl(s, a))$ is strictly monotonically increasing for $0 \leq dl(s, a) \leq pm(s, a)$ and is strictly monotonically decreasing for $pm(s, a) \leq dl(s, a) \leq 1$.*

The monotonicity restrictions also ensure that agents who deliver *close* to their promise are considered to be more trustworthy than agents who delivered far from their promise.

Our second desired property is that agents who contribute more than they promised should be considered to be more trustworthy than those who delivered less than what they promised, when they both deviate by ϵ.

Trust Property 2. *For any promise $pm(s,a)$ and for any $\epsilon > 0$*

$$Tr(pm(s,a), pm(s,a)+\epsilon) > Tr(pm(s,a), pm(s,a)-\epsilon).$$

The third property states that if two agents both over- or under-deliver on their promise, then the one who delivered services closest to its promised level is more trustworthy.

Trust Property 3. *Assume agents a and a' have promised $pm(s,a)$ and $pm(s,a')$ respectively, and assume that $dl(s,a) = dl(s,a') = d$. If $d \geq pm(s,a) > pm(s,a')$ then $Tr(pm(s,a),d) > Tr(pm(s,a'),d)$. If $pm(s,a) > pm(s,a') \geq d$ then $Tr(pm(s,a),d) < Tr(pm(s,a'),d)$.*

Our fourth property states that if two agents over-deliver on their promise by the same amount ϵ then the one who promised more is more trustworthy, and if both under-deliver their promise by the same amount ϵ then the one who promised less is more trustworthy, since the expectations about the agent were less to start with.

Trust Property 4. *Assume that agent a made promise $pm(s,a)$. Then for any $\epsilon > 0$ and for any $p' < pm(s,a)$*

$$Tr(pm(s,a), pm(s,a)+\epsilon) > Tr(p', p'+\epsilon)$$

and

$$Tr(pm(s,a), pm(s,a)-\epsilon) < Tr(p', p'-\epsilon).$$

6.2.4 Instantiating the Trust Model

In the previous section we proposed a set of properties we believe a promise-based trust model should have. In this section we provide a particular instantiation of the trust model which satisfies our properties. We note that other families of functions may also work, and that the application in which our model is used would play a role in determining what an appropriate set of functions would be. To simplify our notation, in this section we refer only to agent a and service s, and let $p = pm(s,a)$ and $d = dl(s,a)$.

We first propose a reliability function

$$RL(p,d) = \begin{cases} 0 & \text{if } d\text{=}0 \\ 1 - |d-p| & \text{otherwise} \end{cases} \tag{6.1}$$

The sole requirement we had for the reliability function was that it had to be monotonically increasing for $d \in [0,p]$, monotonically decreasing for $d \in [p,1]$, and $RL(p,0) = 0$. It is straightforward to verify that $RL(p,d)$ satisfies this requirement.

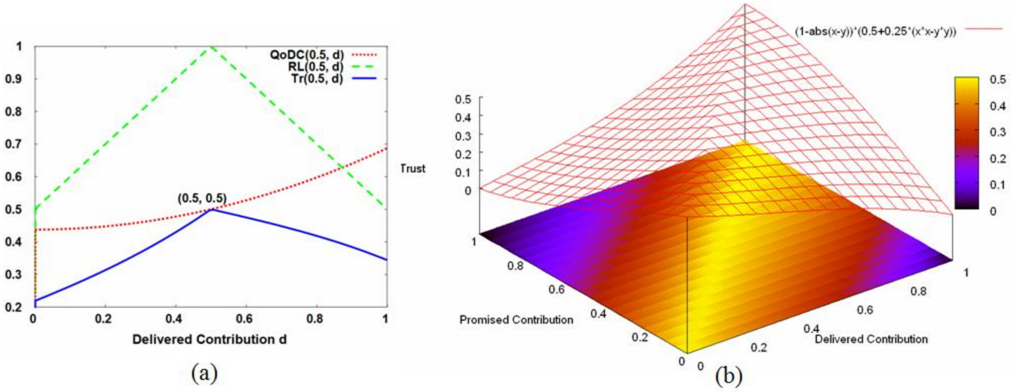


Figure 6.1: (a) Instances of RL, QoC and Tr functions for $n = 2$ and $p = 0.5$ (b) $Tr = \frac{1}{n} + \frac{d^n - p^n}{n^2}$ for $p, d \in [0, 1]$ and $n = 2$.

We propose the following family of functions from which the QoC function could be selected

$$QoC(p,d) = \begin{cases} 0 & \text{if } d{=}0 \\ \frac{1}{n} + \frac{(d^n - p^n)}{n^2} & \text{otherwise} \end{cases} \tag{6.2}$$

where $n \in \mathbb{N}$, $n > 1$. We now show that this family of functions satisfies the properties we outlined for the QoC function.

QoC Property 1: Note that for any p, $0 \leq p \leq 1$ and $0 < d \leq 1$ and $n > 1$

$$\frac{d^n - p^n}{n^2} \geq -\frac{1}{n^2} \Leftrightarrow \frac{1}{n} + \frac{d^n - p^n}{n^2} > \frac{1}{n} - \frac{1}{n^2} > 0$$

Therefore, function (6.2) is always positive. For the monotonicity, it is sufficient to show that

$$\frac{\partial QoC(p,d)}{\partial d} > 0$$

The partial derivative of $QoC(p, d)$ with respect to d is

$$\frac{\partial QoC(p,d)}{\partial d} = \frac{d^{n-1}}{n} > 0$$

since $d > 0$ and $n > 1$. Thus, *QoC Property 1* is satisfied.

QoC Property 2: It is sufficient to show that $QoC(p, p + \epsilon)$ is strictly monotonically increasing in p. This can be done by considering the partial derivative of QoC with respect to p. More specifically, we just have to show that it is positive. Thus

$$\frac{\partial QoC(p, p+\epsilon)}{\partial p} = \frac{(p+\epsilon)^{n-1} - p^{n-1}}{n} > 0 \tag{6.3}$$

which is positive since the function $\phi(x) = x^n$ is strictly monotonically increasing for $\forall x > 0$ and $n > 0$.

In Equation 6.1 we defined the trust function using a generic operation, $\diamond$. We now propose to replace that operation with *multiplication* resulting in

$$\begin{aligned} Tr(p,d) &= RL(p,d) \cdot QoC(p,d) \\ &= (1 - |d-p|)(\frac{1}{n} + \frac{d^n - p^n}{n^2}). \end{aligned}$$

We now show that this trust function satisfies the three properties outlined in Section 6.2.3.

Trust Property 1: Consider first the case $d \in [0, p]$. We have

$$\begin{aligned} RL(p,d) &= 1 - p + d \\ QoC(p,d) &= \frac{1}{n} + \frac{1}{n^2}(d^n - p^n). \end{aligned} \tag{6.4}$$

Since $RL(p,d)$ and $QoC(p,d)$ are both positive and strictly monotonically increasing in $[0, p]$, the function

$$Tr(p,d) = RL(p,d) \cdot QoC(p,d) \tag{6.5}$$

is also strictly monotonically increasing.

Now, consider the case where $d \in [p, 1]$. It is sufficient to show that $\frac{\partial Tr(p,d)}{\partial d} < 0$. We have

$$\begin{aligned} Tr(p,d) &= (1 - d + p) * (\frac{1}{n} + \frac{1}{n^2}(d^n - p^n)) \Leftrightarrow \\ Tr(p,d) &= \frac{1}{n} + \frac{d^n}{n^2} - \frac{p^n}{n^2} - \frac{d}{n} - \frac{d^{n+1}}{n^2} + \frac{d * p^n}{n^2} + \\ &+ \frac{p}{n} + \frac{pd^n}{n^2} - \frac{p^{n+1}}{n^2} \end{aligned} \tag{6.6}$$

so

$$\frac{\partial Tr(p,d)}{\partial d} = \frac{p^n - d^n}{n^2} + \frac{d^{n-1}}{n}(p-d) + \frac{d^{n-1} - 1}{n} \tag{6.7}$$

Since $d > p$, $d \leq 1$, $n \in \mathbb{N}$ and $n > 1$ we have

$$\frac{p^n - d^n}{n^2} < 0 \textbf{ and } \frac{d^{n-1}}{n}(p - d) < 0 \textbf{ and } \frac{d^{n-1} - 1}{n} \leq 0 \tag{6.8}$$

Thus $\frac{\partial Tr(p,d)}{\partial d} < 0$.

Trust Property 2: Let $d = \epsilon + p$ and $d' = p - \epsilon$. Now, we wish to show that $Tr(p, d) > Tr(p, d')$ which is equivalent to

$$QoC(p, d) \cdot RL(p, d) > QoC(p, d') \cdot R(p, d'). \tag{6.9}$$

Since $RL(p, d) = RL(p, d') = 1 - \epsilon$, it is sufficient to show that $QoC(p, d) > QoC(p, d')$. This is true since QoC is strictly monotonically increasing with respect to the delivered contribution (QoC Property 1) and $d > d'$. Thus, *Trust Property 2* is satisfied.

Trust Property 3: For this property it is sufficient to show that the function $Tr(p, d)$ is monotonically increasing in $[0, d)$ and strictly monotonically decreasing in $(d, 1]$ with respect to the promised contribution, p. For $p \in [0, d)$

$$\begin{aligned} Tr(p,d) &= (\frac{1}{n} + \frac{d^n - p^n}{n})(1 - d + p) \Rightarrow \\ \frac{\partial Tr(p,d)}{\partial p} &= \frac{1}{n} - \frac{p^{n-1}}{n^2} + \frac{dp^{n-1}}{n^2} \\ &+ \frac{d^n}{n^2} - \frac{(n+1)p^n}{n^2} \\ &= (\frac{1 - p^{n-1}}{n}) + \frac{p^{n-1}(d - p)}{n} + \frac{d^n - p^n}{n^2} > 0 \end{aligned} \tag{6.10}$$

since $1 \geq d > p$ and $n > 1$. Thus, the function $Tr(p, d)$ is monotonically increasing when $p \in [0, d)$.

Similarly, for $p \in (d, 1]$

$$\begin{aligned} Tr(p,d) &= (\frac{1}{n} + \frac{d^n - p^n}{n^2})(1 + d - p) \Rightarrow \\ \frac{\partial Tr(p,d)}{\partial p} &= -\frac{1}{n} - \frac{d^n}{n^2} - \frac{p^{n-1}}{n} - \frac{dp^{n-1}}{n} \\ &+ \frac{p^n}{n} + \frac{p^n}{n^2} \Rightarrow \\ &= \frac{p^n - 1}{n} - \frac{p^{n-1}}{n}(\frac{p}{n} - 1) - \frac{d^n}{n^2} - \frac{dp^{n-1}}{n} < 0 \end{aligned} \tag{6.11}$$

since $1 \geq p > d$ and $n > 1$. Thus, the function $Tr(p, d)$ is monotonically decreasing when $p \in (d, 1]$.

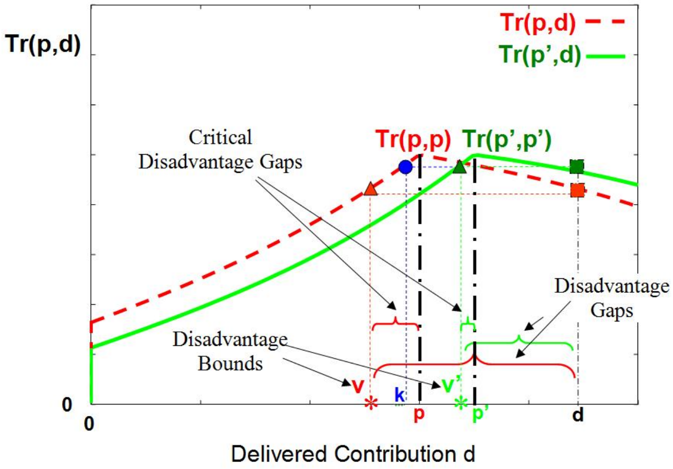


Figure 6.2: Examples of the Disadvantage Bound, Gap and Critical Gap for p and p' where $p' > p$.

Trust Property 4: Since $RL(p, p + \epsilon) = 1 - |\epsilon|$, then from QoC Property 2 the result follows.

Figure 6.1 depicts examples of RL, QoC, and Tr. As we can see in Figure 6.1(a), the QoC function monotonically increases as the delivered contribution d increases (QoC Property 1). As for the trust function, Tr, when the delivered contribution is less than or equal to the promise then Tr is monotonically increasing. Otherwise, the trust function decreases monotonically (Trust Property 1). Furthermore, we can observe that the rate in which it increases is faster than the rate in which it decreases. Thus, it is easy to see that Trust Property 2 is satisfied. The QoC Property 2 and Trust Property 3 cannot be observed in Figure 6.1(a) since they focus on the dimension of the promised contribution p. Figure 6.1(b) depicts the Tr function for $n = 2$ and $p, d \in [0, 1]$.

6.2.5 The Disadvantage Gap

One interesting property of our trust model is that it provides incentives so that agents will not promise less than what they can actually deliver. In this section we will prove that if an agent knows it can deliver some level of service $dl(s, a)$, then it puts itself at a disadvantage (i.e., reduces its trust value) by promising (and then delivering) some lesser contribution. Before proving this, we need to introduce two new concepts: the *disadvantage bound* and the *disadvantage gap*.

Definition 13. *Let $pm(s, a)$ be the promised contribution of agent a, and let $dl(s, a)$*

be the delivered contribution. Assume that $dl(s,a) > pm(s,a)$. That is, the agent contributed more than it promised. The Disadvantage Bound *is the minimum contribution, $v = v(pm(s,a), dl(s,a))$, the agent could have delivered such that*

$$Tr(pm(s,a), v) = Tr(pm(s,a), dl(s,a)).$$

Definition 14. *Let $pm(s,a)$ be the promised contribution of agent a. For any delivered contribution $dl(s,a) > pm(s,a)$ and associated Disadvantage Bound $v = v(pm(s,a), dl(s,a))$, the* Disadvantage Gap *is the interval*

$$[v, dl(s,a)].$$

Consider an agent who promised $pm(s,a)$ and then delivered $dl(s,a) > pm(s,a)$. The Disadvantage Gap is the region of contributions that some agent a' who promised $pm(s,a') = pm(s,a)$ could make (even under-contributing such that $dl(s,a') < pm(s,a')$) and yet still have the same or higher trust value as agent a.

Definition 15. *Let $pm(s,a)$ be the promised contribution of agent a. For any delivered contribution $dl(s,a) > pm(s,a)$ and associated Disadvantage Bound $v = v(pm(s,a), dl(s,a))$, the* Critical Disadvantage Gap *is the interval*

$$[v, pm(s,a)].$$

The *gaps* represent ranges of contributions which result in the same or greater trust values than the trust value currently received by the agent who promised and delivered $pm(s,a)$ and $dl(s,a)$. In particular, the intervals are the ranges in which another agent, making the same promise, could under-deliver by and still be assigned a greater trust value. Figures 6.3 and 6.4 illustrate the disadvantage bounds and the critical disadvantage gaps: i) for three different promises and different deliverables and ii) for $p = 0.7$ and different values of n.

What we show in Theorem 2 is that if an agent can deliver $dl(s,a)$, then it is to its advantage to make a promise which is equal to $dl(s,a)$ or possibly a little less, as opposed to significantly under-promising. In particular, we will show that as $pm(s,a)$ approaches $dl(s,a)$ the range for which another agent could under-deliver and still receive a higher trust rating decreases. Thus, an agent is better off, from a trust perspective, to try to promise close to, if not exactly, what it can actually deliver. Note that Theorem 2 also implies Corollary 1.

Theorem 2. *Let $Tr : [0,1] \times [0,1] \mapsto \mathbb{R}^+$ be a function which satisfies our four trust properties. Let $dl(s,a)$ be the actual delivered contribution of agent a. For promise $pm(s,a) < dl(s,a)$ let $v = v(pm(s,a), dl(s,a))$ be the disadvantage bound. Then, as $pm(s,a)$ approaches $dl(s,a)$, the size of the Critical Disadvantage Gap $[v, pm(s,a)]$ decreases.*

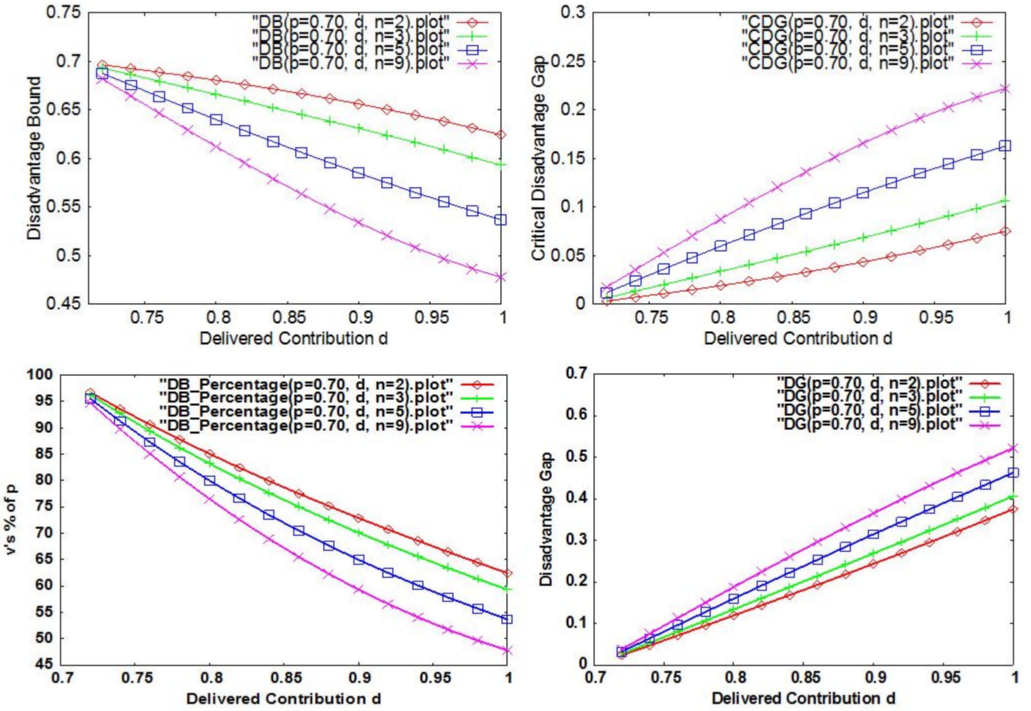


Figure 6.3: Disadvantage Bound, Disadvantage Bound percentage over promise $p = 0.7$, and Disadvantage Gap and Critical Gap diagrams $\forall d \in (p, 1]$ given $p = 0.7$ and $n = \{2, 3, 5, 9\}$.

Proof. Let $p = pm(s, a)$ and $d = dl(s, a)$. Assume that another promise, p' was made such that $p < p'$. Let $v = v(p, d)$ and $v' = v(p', d)$ be the respective disadvantage bounds. We are required to show that for any p and p' with $p < p' < d$ it is the case that $p - v > p' - v'$.

Since v and v' are the disadvantage bounds for p and p' then by definition, $Tr(p, v) = Tr(p, d)$ and $Tr(p', v') = Tr(p', d)$. Since $d > p' > p$ then according to Trust Property 3 we have that $Tr(p', d) > Tr(p, d)$, and so therefore $Tr(p', v') > Tr(p, v)$.

Since $p' - v' > 0$ then by Trust Property 4 we have that $Tr(p, p - (p' - v')) > Tr(p', p' - (p' - v')) = Tr(p', v')$ and thus $Tr(p, p - (p' - v')) > Tr(p, v)$ since $Tr(p', v') > Tr(p, v)$. Therefore, given Trust Property 1 we have that $p - (p' - v') > v$ and thus $p - v > p' - v'$. □

Corollary 1. *Let Tr be a trust function that satisfies Trust Properties 1 and 3, and let $d = dl(s, a)$ be the delivered contribution of agent a. Then, the* Disadvantage Gap *for agent a decreases as the promised contribution $pm(s, a)$ increases, for $pm(s, a) < d$.*

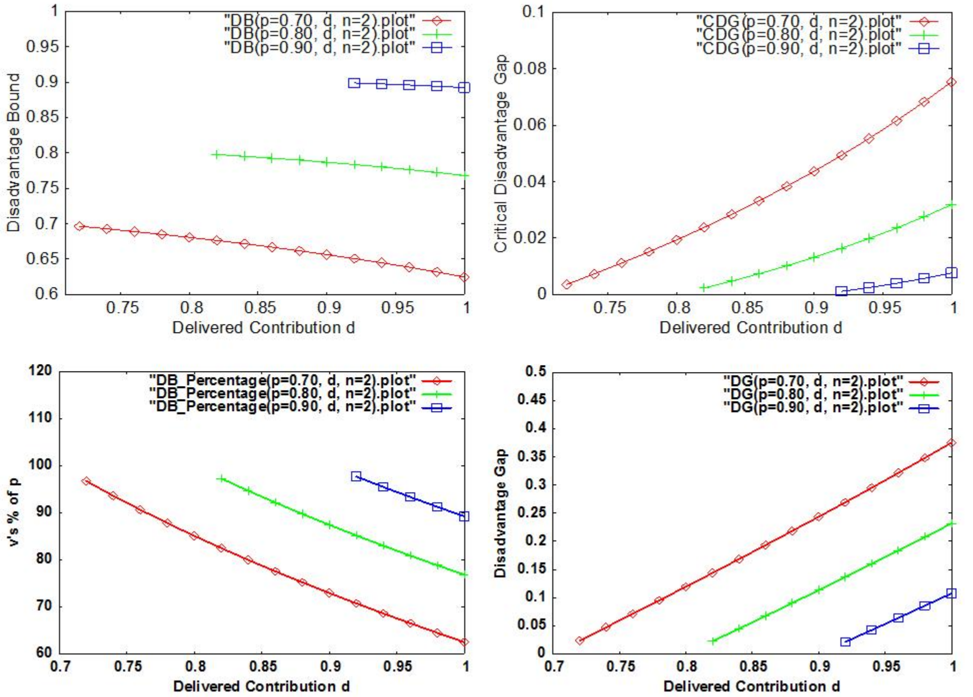


Figure 6.4: Disadvantage Bound, Disadvantage Bound percentage over promise p, and Disadvantage Gap and Critical Gap diagrams $\forall d \in (p, 1]$ and $p = \{0.7, 0.8, 0.9\}$ given $n = 2$.

6.2.6 Allowing for Changing Circumstances

The main goal of this chapter is to develop an incentive-compatible trust model, where agents who make good promises, and then deliver on these promises, are regarded as trustworthy. We have achieved this since we need only to prove that $Tr(pm(s,a), dl(s,a))$ is uniquely maximized when $pm(s,a) = dl(s,a)$. This is achieved through *Trust Property 2* and thus any trust function which satisfies this property is incentive compatible.

However, there is one unintended side-effect of our trust model as described to date. Consider an agent who, when asked to make a promise about its contribution, is uncertain as to what it will be able to actually deliver when called upon to do so. To be safe, the agent may promise *less* than it may actually be able to deliver. For example, assume that agent a has promised $pm(s,a)$, but then finds that it is capable of delivering $dl(s,a) = pm(s,a) + \epsilon$. If the agent actually delivers this service at this new level, then while its QoC score would increase compared to if it delivered what it promised, its overall trustworthiness would decrease. Thus, the agent may decide not to deliver the extra ϵ of service. In a number of cases over-delivering might be undesirable from the community's perspective since the community must make plans based on the promised services. This is due to the fact that there are cases when the update on the agent's promises might circumscribe the planning process of the community.

To overcome the problem outlined above, we would like to allow agents the ability to *update* their promises, so that once they are better able to predict their contribution capabilities, they can inform the community without being penalized. On the other hand, however, we need to ensure that such a mechanism is not abused by agents in an attempt to manipulate the system and improve their trust scores, by always initially making small promises.

The main idea is that if the delivered contribution of an agent is different than its promised contribution but it is done during a certain time period in which any change will not impact the community in any key way, then the trustworthiness of the agent should not be penalized. Instead, we propose that both the QoC function and the trust value, Tr should be computed using $pm(s, a) = dl(s, a)$. If the latter time period expires then the trustworthiness of the agent should get penalized except when certain conditions apply. In the rest of the section we enumerate these conditions but first we provide some necessary definitions.

Definition 16. *Given a set of alternative allocations of goods or outcomes for a set of agents, a change from one allocation to another that can make at least one agent better off without resulting in any other agent being worse off is called a* Pareto Improvement*.*

Definition 17. *An allocation is* Pareto Optimal *when no further Pareto improvements can be made.*

In our setting our definition of *worse off* is as follows:

Definition 18. *An agent a_i is* worse off *with respect to the contribution of another agent a_j, if a_j's exact reporting of its contribution would have resulted in a_i acquiring a higher utility.*

For example, consider two agents a_1 and a_2. Assume that a_2 promised a contribution $pm(s, 2) = p$ but provided a contribution p'. The agent a_1 is worse off with respect to the contribution of a_2 if the reporting p' of a_2's contribution would have resulted in a higher utility for a_1.

We propose the following protocol to deal with changing circumstances. First, upon making its initial promise, we allow for some time period in which the agent may *update* its promise with no penalty. After this agreed-upon time period has passed, then the agent is only allowed to make updates if one of the two following condition holds:

- The updated delivered contribution results in a *Pareto Improvement*.

- The allocation of the benefits the agents would have received with respect to their contributions was Pareto optimal *and* satisfies the following two requirements:
  1. results in a Pareto optimal allocation,
  2. the updated promised contribution is equal or greater than the average promised contribution. [1]

For each of these cases the agent's trustworthiness should not get penalized and its trust score will be computed as $Tr(d, d)$ where d is the delivered contribution. In any other case, the agent's trust score is computed as $Tr(p, d)$ based on its initial contribution p. We believe that this approach allows an appropriate level of flexibility for agents, but at the same time protects the community of the agents from malicious strategic behaviour.

Summarizing, in order for an agent to avoid reducing its trust value by delivering a different contribution than it promised, it must either update its promise during the predefined time period, or hope that its delivered contribution is a Pareto Improvement or results in the same Pareto Optimal allocation.

6.3 Examples

In this section we provide some examples of scenarios where our trust function would be used and the numerical values of the trustworthiness of each participant for each scenario. The instances of the functions we use are the following:

$$RL(p, d) = 1 - |d - p| \tag{6.12}$$

and

$$QoC(p, d) = \frac{1}{2} + \frac{d^2 - p^2}{4} \tag{6.13}$$

and

$$\begin{aligned} Tr(p, d) &= RL(p, d) \cdot QoC(p, d) \\ &= (1 - |d - p|)(\frac{1}{2} + \frac{d^2 - p^2}{4}) \end{aligned} \tag{6.14}$$

Note that we will only consider the normalized value of the function Tr in the interval $[0, 1]$.

[1] This condition aims to prevent a domino effect from taking place.

6.3.1 Over-delivery

Consider a set of agents $\{a_1, a_2, a_3, a_4\}$ that are interested in selling a set of tickets through an online electronic company. Each agent has to declare the number of tickets it has to sell. For each ticket they sell they have to pay two fees: i) posting fee PF and ii) shipping fee SF. The company offers the following plans:

- **Plan A:** If *number of tickets* < 10 then $PF = \$1.5$ and regular SF
- **Plan B:** If $10 \leq$ *number of tickets* < 25 then $PF = \$1.2$ and regular SF
- **Plan C:** If $25 \leq$ *number of tickets* < 50 then $PF = \$0.8$ and regular SF
- **Plan D:** If $50 \leq$ *number of tickets* then $PF = \$0.4$ and reduced SF

Agents can form groups in order to achieve their participation in a better plan. Once a group commits to a number of tickets it cannot decrease it without the trustworthiness of the group getting penalized. Let assume that the agents a_1, a_2, a_3 and a_4 can offer 15, 20, 15 and 5 tickets, respectively. The agents a_1, a_2, a_3 and a_4 form a group and thus the total number of tickets they can offer is 55. With 45 tickets in total the group of the agents can participate in *Plan C*. Assume now that a_1, a_2, and a_3 were truthful but a_4 was not.

The agent a_4 played strategically and only reported 5 from the 11 tickets it owns. The agent a_4 reserved the rest of the 6 tickets to contribute to another group of agents. Assume now that the other deal a_4 was trying to participate failed and a_4 is interested now in selling the remaining 6 tickets through the *Plan C* pretending that it honestly did not know about these extra 6 tickets. If the agents cannot update their offer so they can access *Plan D* only the agent a_4 is benefitted. In this case by using our proposed Trust function the trustworthiness of the agent a_4 inside the group is not maximized (i.e., 1) due to the possibility that the agent a_4 was intentionally dishonest. However, its QoC value can still reveal that the non-perfect trustworthiness is the result of over delivering.

agent a	$pm(s,a)$	$dl(s,a)$	$Tr(s,a)$	$QoC(s,a)$	$Rl(s,a)$
a_1	$\frac{10}{50} = 0.2$	$\frac{10}{50} = 0.2$	1	0.5	1
a_2	$\frac{20}{50} = 0.4$	$\frac{20}{50} = 0.4$	1	0.5	1
a_3	$\frac{15}{50} = 0.3$	$\frac{15}{50} = 0.3$	1	0.5	1
a_4	$\frac{5}{50} = 0.1$	$\frac{11}{50} = 0.22$	0.897	0.510	0.88

Table 6.1: The Tr, QoC, and Rl values for E-market Over-Delivery Example

If a_4 was honest from the beginning and had reported all of its 11 tickets the agents could participate in *Plan D* instead of *Plan C*. Thus, they would have all achieved a better deal. Table 6.1 depicts the normalized trustworthiness Tr value, the quality of contribution QoC

value and the reliability Rl value for each agent in $\{a_1, a_2, a_3, a_4\}$ when using the functions (6.14), (6.13), and (6.12), respectively. In particular, the trustworthiness of the agents a_1, a_2, a_3, and a_4 is 1, 1, 1, and 0.897, respectively.

6.3.2 Under-delivery

Consider an electronic market place that is in urgent need of three very popular items: a) a new game console, b) a new book and c) tickets for a concert. The site requests from the participants to contribute any of the above items. In particular, a contribution of 10 game consoles or of 20 books or of 6 tickets with seats next to each other would be characterized as *sufficient*. Consider now 6 agents such that:

- a_1: promised *8 game consoles* but delivered *7 game consoles*
- a_2: promised *5 books* but delivered *4 books*
- a_3: promised *3 tickets* and delivered *3 tickets*
- a_4: promised *10 books* but delivered *9 books*
- a_5: promised *4 tickets* but delivered *3 tickets*
- a_6: promised *2 tickets* but delivered *1 ticket*

We can calculate $pm(s, a)$ based on the number of items the agent contributed divided by the number of items the market considers as a *sufficient* contribution. If the contribution exceeds the latter number we consider a contribution equal to this number. Table 6.2 depicts the normalized trustworthiness Tr value, the quality of contribution QoC value and the reliability Rl value for each agent in $\{a_1, a_2, a_3, a_4, a_5, a_6\}$ when using the functions (6.14), (6.13), and (6.12), respectively. In particular, the trustworthiness of the agents a_1, a_2, a_3, a_4, a_5, and a_6, are 0.833, 0.939, 1 , 0.927, 0.762 and 0.806, respectively.

agent a	$pm(s, a)$	$dl(s, a)$	$\epsilon = dl(s, a) - pm(s, a)$	$Tr(s, a)$	$QoC(s, a)$	$Rl(s, a)$
a_1	$\frac{8}{10} = 0.8$	$\frac{7}{10} = 0.7$	-0.1	0.833	0.463	0.9
a_2	$\frac{5}{20} = 0.25$	$\frac{4}{20} = 0.2$	-0.05	0.939	0.494	0.95
a_3	$\frac{3}{6} = 0.5$	$\frac{3}{6} = 0.5$	0	1	0.5	1
a_4	$\frac{10}{20} = 0.5$	$\frac{9}{20} = 0.45$	-0.05	0.927	0.49	0.95
a_5	$\frac{4}{6} = 0.66$	$\frac{3}{6} = 0.5$	-0.16	0.762	0.454	0.84
a_6	$\frac{2}{6} = 0.33$	$\frac{1}{6} = 0.17$	-0.16	0.806	0.48	0.84

Table 6.2: The Tr, QoC, and Rl values for E-market Under-Delivery Example

As we can see from Table 6.2 the agents a_5 and a_6 deviated both by -0.16 (i.e., 1 ticket out of 6 tickets) but the agent a_5 receives a higher trustworthiness than the agent a_6. In general, the higher the number of the tickets that refer to seats next to each other the less likely is to find a seller than can provide these tickets. Thus, a lie in a higher range should result in a higher penalty. For instance, assume that the market place had committed to provide to a buyer four tickets that refer to 4 seats next to each other if a_5 ends up only delivering 3 tickets the marketplace will not be able to satisfy this request resulting to the dissatisfaction of the buyer.

Similar is the case of the agents a_2 and a_4 that deviated by $\epsilon = -0.05$ from their initial promise (i.e., delivered one book out of twenty books). The agent a_4 that gave the higher promise will receive a lower score. Similarly to the case of the tickets, as the likelihood of finding sellers to offer this number of books decreases. For instance, consider a buyer that is interested to purchase 10 books and that only a_2 was offering to sell 10 books. If a_2 only ends up providing 9 the market-place should find the extra book from another seller a'. Having to ship the books to the buyer from two different sellers will result in an additional shipping cost to either the buyer or the marketplace.

Note that in our example we used the same instantiation of the Tr, QoC, and RL functions. However, different instantiations could be used with respect to the importance of each item (i.e., different for the books, for the tickets and so on). Furthermore, the decision of the most proper instantiations may differ from setting to setting and is subject to the conditions that are more desirable.

6.4 Experiments

In this section we present an empirical evaluation of the over delivery component of our proposed model (experiment 1) in addition to an effort to study the possibility of extending a probabilistic-based trust model to approximate our proposed trust properties (experiment 2). We focused on the over-delivery case since this is the case that at a first glance appears to be counter-intuitive.

6.4.1 Experiment 1

To the best of our knowledge there is no trust model that was designed to consider both under-delivery and over-delivery of contributions of goods or services. For this reason we focused on comparing our model with a probabilistic model in a setting where the probabilistic model can reach its optimal performace. Due to symmetry we only focused on the case where over-delivery takes place. We chose the over-delivery case because this also demonstrates the negative impact over-delivery can have.

In particular, the goal of experiment 1 is twofold:

1. to examine the extent a deployment of a trust model can achieve an increase in the utility (i.e., revenue) for truthful agents in settings where the untruthful agents over deliver their contribution, and
2. to examine the degree the instantiation of the promise-based trust model that we provided in section 6.2.3 can compete with a standard probabilistic model [25].

Experiment 1a

Experiment 1a examines the degree the instantiation of the promise-based trust model that we provided in section 6.2.3 compares with the probabilistic model [25] in settings where the probabilistic can reach its optimal performance.

Scenario: The experiment follows a similar scenario to the one in Example 1. We have a group G of agents which are interested in participating in a plan that provides a certain deal. The plan the group G can access depends on the budget $B(G)$ of the group. The higher the budget the better deal the group G can access. In order to increase the likelihood of participating in a good plan the agents in G wish to collaborate with a set of agents $S \subseteq A_{UT}$, where A_{UT} is the set of manipulating agents. The agents in A_{UT} appear to play strategically and under-declare their contributions. More specifically, when an agent in $a \in A_{UT}$ promises a contribution $pm(a)$ it ends up delivering a contribution equal to $pm(a) + \epsilon$ for some $\epsilon > 0$. In order to ensure that $pm(a) + \epsilon \leq 1$ we consider that for the case of the untruthful agents $pm(a) \in [0, 1 - \epsilon]$.

The revenue $R(G)$ of the group G is equal to the deal $o(B(G) + B(S))$ the group G achieved, after collaborating with the agents in S, multiplied by the group's budget $B(G)$. In particular

$$R(G) = o(B(G) + B(S)) \cdot B(G) \tag{6.15}$$

where $B(G)$ and $B(S)$ is the budget of group G and S, respectively. Each group G participates in a number of transactions that make an *epoch*. The goal of the group G

is to maximize the revenue it accumulates in each *epoch*. We consider that during each transaction only a subset L of A_{UT}, where $L \supseteq S$, can contribute to the budget of the group G. Since which plan the group can enter depends also on the contributions of the agents in S it is important to decide how to select the set of agents S from the set L. For our experiment for determining the set S we consider the following two methods.

- **Method A:** S consists of the agents in L with the $|S|$ highest promised contributions.
- **Method B:** S is determined as follows:
  - 1^{st} Step: Add in S the agents in L with the $|S|$ highest promised contributions.
  - 2^{nd} Step: Find the plan P the group G can access based on its budget $B(G) + B(S)$.
  - 3^{rd} Step: Let a be the agent with the highest promised contribution in L. **If** there is an agent a' in $L - S$ such that
    - $TR^t(a') > TR^t(a)$ **and**[2]
    - the group $S' \leftarrow S - \{a\} \cup \{a'\}$ has a budget $B(G) + B(S')$ that gives access to the same plan than the one the group G can access.[3]

    then replace a with a' in S, and repeat the procedure for the agent a with the second highest promised contribution in S and so on until the agent with the $|S|$-th highest contribution in S is reached.
  - 4^{th} Step: Repeat Step 3 until no swapping of agents between S and $L - S$ takes place.

Essentially, Method B tries to find the set S with the highest trustworthiness that could achieve a plan equal to the best plan G can achieve given that it can only select $|S|$ agents from the set L.

We compared 3 different settings:

Setting 1: **No Trust:** In this setting we consider that there is no trust mechanism applied. Thus, the agents in A_{UT} always under-declare their contributions while the selection of the group G is done through *method A*.

Trust-based Settings. In the following settings the selection of the group G is done through *method B*. We assume that each untruthful agent can observe and estimate the trustworthiness of a set of untruthful agents, which we refer to as the *horizon*. The agents in A_{UT} will only tell the truth if

[2] t is the time the group is formed.

[3] The plan that is selected by considering the agents in L with the m highest promised contributions is guaranteed to be the best plan that the group G can access.

1. *their overall trustworthiness TR is not the maximum possible (i.e., 1) and it is below the average overall trustworthiness of half of the agents in their* horizon **or**

2. *they have participated in the list L but not in S more than a number of times which we will refer to as the "tolerance" of the agent.*

The trustworthiness $TR(a)$ of each agent a in S is updated only at the end of every epoch.

- Setting 2: **Promise-based**: The overall trustworthiness of an agent a in epoch t is computed at the end of epoch $t-1$ as follows:

$$TR^t(a) = TR_m^{t-1}(a) \tag{6.16}$$

  where m is the number of interactions during epoch t and

$$TR_i^{t-1}(a) = \gamma Tr_i^{t-1}(pm_i^{t-1}(a), dl_i^{t-1}(a)) + (1-\gamma) TR_{i-1}^{t-1}(a) \tag{6.17}$$

  where $TR_0^{t-1} = TR^{t-1}(a)$, $Tr_i^{t-1}(pm_i^{t-1}(a), dl_i^{t-1}(a))$, $pm_i^{t-1}(a)$ and $dl_i^{t-1}(a)$ are the trustworthiness, the promised contribution and the delivered contribution of the agent a during the i-th transaction (or iteration) in epoch $t-1$, respectively. The parameter γ defines the extent the new trust rating influences the overall trustworthiness.

  The $Tr_i^{t-1}(pm_i^{t-1}(a), dl_i^{t-1}(a))$ is computed based on our proposed trust function

$$Tr_i^{t-1}(pm_i^{t-1}(a), dl_i^{t-1}(a)) = (\frac{1}{2} + \frac{dl_i^{t-1}(a)^2 - pm_i^{t-1}(a)^2}{4})(1 + dl_i^{t-1}(a) - pm_i^{t-1}(a)) \tag{6.18}$$

  and is normalized in $[0,1]$.

- Setting 3: **Probabilistic.** The overall trustworthiness of an agent a in epoch t is computed at the end of epoch $t-1$ as follows

$$TR^t(a) = \gamma Tr^t(a) + (1-\gamma) TR^{t-1}(a) \tag{6.19}$$

  where

$$Tr^t(a) = \frac{b}{b+c} \tag{6.20}$$

  where b and c are the number of times during epoch $t-1$ the agent proved to be truthful and untruthful, respectively.

For this experiment we consider $\gamma = 0.5$. By setting $\gamma = 0.5$ we weight equally the trustworthiness of each epoch and the past overall trustworthiness when computing the new overall trustworthiness of an agent. The appropriate value of γ depends on the behaviour of the agents. A reinforcement learning technique could be applied in order to determine the optimal γ value with respect to the behaviour of the agents at each time. We chose $\gamma = 0.5$ since by setting a value of $\gamma < 0.5$ an agent can build a high overall trustworthiness TR and then counting on the fact that it will take a while until a deceptive behaviour is reflected on its trustworthiness TR, it may start to significantly under-declare its contributions (the smaller the γ the longer it will take). In the case of $\gamma > 0.5$ an agent might continuously under-declare its contribution and occasionally provide truthful reports in order boost its overall trustworthiness.

The budget of the group G is selected from the range $[0, 8]$ uniformly at random, $|A_{UT}|$ is set to 500, the horizon to 10 which is 2% of the agents in A_{UT}, $|S|$ to 4, γ to 0.5, $|L|$ to 50 which is the 10% of the agents in A_{UT} and the $Tolerance$ to 40 which corresponds to 2 epochs assuming that an agent was selected in L, but not in S, in all the iterations of 2 epochs. In each case we assumed that each agent in A_{UT} under-declares by the same amount ϵ, where $\epsilon \in \{i \cdot 0.1 | i = 1, ..., 9\}$. Depending on its $B(G)$ and the budget the agents in S can contribute, one of the following plans will get selected

- **Plan A:** $0 < B(G) + B(S) \leq 1$ Offer: $o = \$100$
- **Plan B:** $1 < B(G) + B(S) \leq 3$ Offer: $o = \$600$
- **Plan C:** $3 < B(G) + B(S) \leq 5$ Offer: $o = \$800$
- **Plan D:** $5 < B(G) + B(S) \leq 7$ Offer: $o = \$1100$
- **Plan E:** $7 < B(G) + B(S)$ Offer: $o = \$1500$

We would like to stress that the above case is a representative of the numerous tests we performed using a large number of different combinations of the values of the above parameters.

For all $\epsilon \in \{i \cdot 0.1 | i = 1, ..., 9\}$ we computed:

1. the average maximum cumulative revenue per epoch the group G would have enjoyed if no agent was untruthful.
2. the average revenue of the group G per epoch for each setting.
3. the percentage of the average cumulative revenue with respect to the average maximum cumulative revenue of the group G per epoch for each setting.

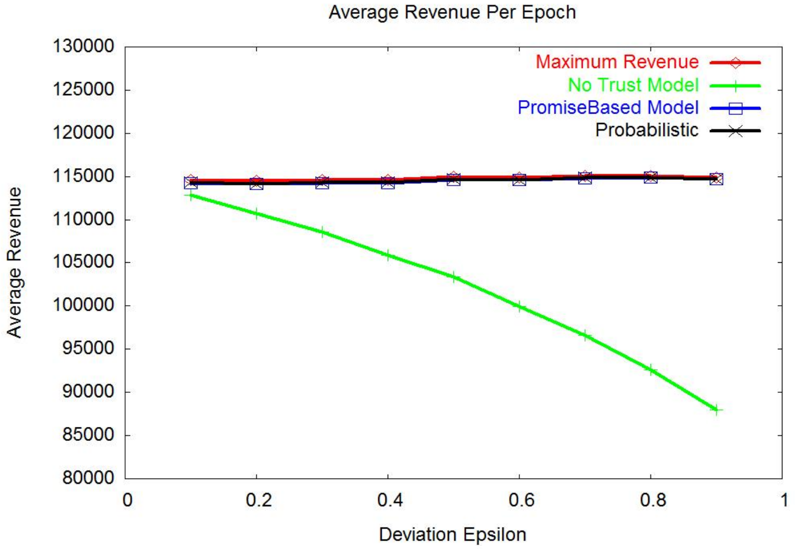


Figure 6.5: Experiment 1a - Average revenue per epoch.

Results - Experiment 1a.

For each ϵ we repeated each experiment 100 times and we present the average results in Figures 6.5 to 6.7.

Figure 6.5 depicts the average revenue a group accumulates per epoch for each of the 3 settings over the 100 times we ran the experiment. The maximum revenue represents the maximum average revenue the group can achieve. As we can see, both the probabilistic and the promise-based trust model are close to the maximum average revenue while in the case where no trust model is used the average revenue decreases significantly.

Figure 6.6 presents the percentage of the maximum revenue each setting approaches. As we also saw in Figure 6.5 both the probabilistic and the promise-based trust model are close to 100% of the maximum revenue with the probabilistic doing slightly better than our proposed promise-based approach.

The exact difference of the latter models can be observed in Figure 6.7 where we can see that the probabilistic model marginally outperforms the promise-based model. As we will discuss in the next section the reason we believe this happens is due to the high scores that the particular instantiation assigns. This is also the reason why the difference between the percentages they achieve overall decreases as the deviation ϵ increases.

Statistical Analysis - Experiment 1a.

In order to validate our results we proceeded to do a statistical analysis of the data we

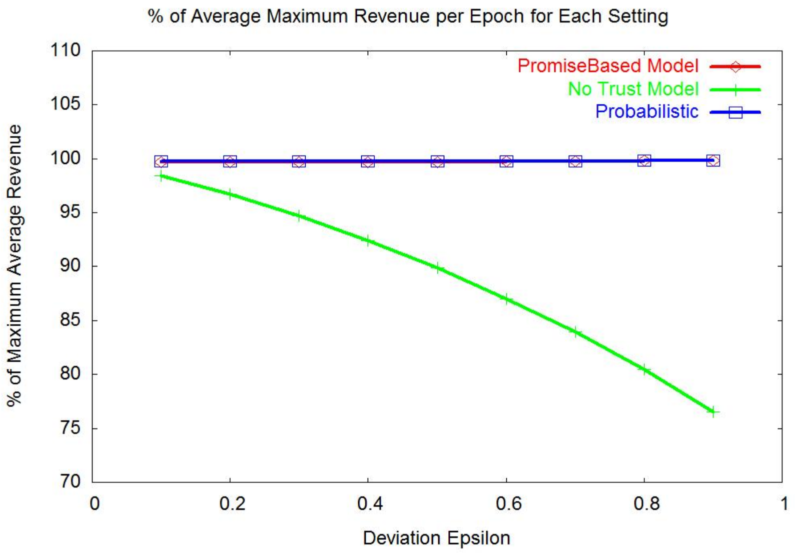


Figure 6.6: Experiment 1a - Percentage of the average maximum revenue per epoch.

ϵ	p-value
0.1	1.35822E-24
0.2	3.9024E-119
0.3	0
0.4	0
0.5	0
0.6	0
0.7	0
0.8	0
0.9	0

Table 6.3: Experiment 1a - p-values for ANOVA Single Value Test.

aggregated. First, given that we are comparing more than two methods we executed an *ANOVA single value test* [45]. The input data is the average revenue per epoch that was computed for each of the 200 epochs of the 100 runs. The p-values for each ϵ are presented in Table 6.3. Given that the p-values for all the ϵ are less than 0.05 we can conclude that for each ϵ there is a statistical significance of the average between the 4 cases we considered.

The next step was to perform paired sample t-tests [82] for the following cases

- No Trust vs Promise-based
- Promise-based vs Probabilistic
- Probabilistic vs Maximum

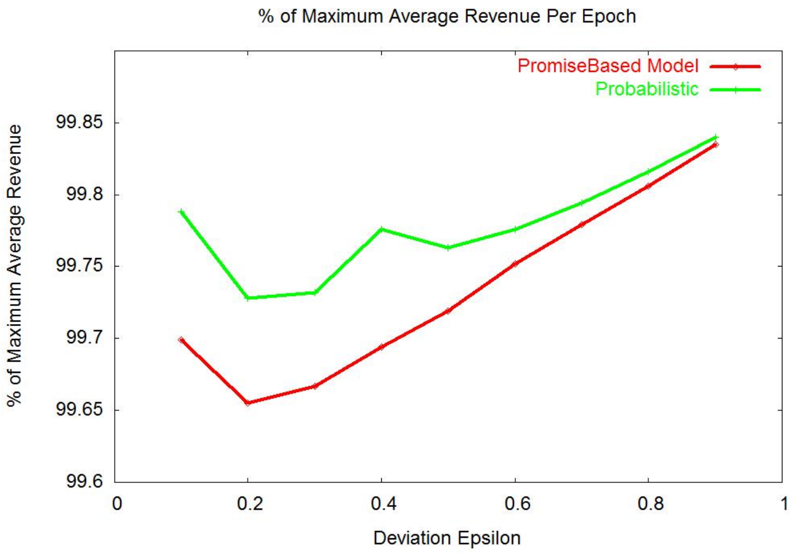


Figure 6.7: Experiment 1a - Percentage of the average maximum revenue per epoch.

ϵ	No Trust vs Promise-based	Promise-based vs Probabilistic	Probabilistic vs Maximum
0.1	0	7.0524E-137	0
0.2	0	1.17357E-84	0
0.3	0	4.40154E-76	0
0.4	0	9.1864E-142	0
0.5	0	2.82523E-47	0
0.6	0	4.05E-20	7.8628E-206
0.7	0	5.00E-12	2.32E-131
0.8	0	6.04681E-08	1.4881E-70
0.9	0	8.50951E-06	3.07501E-41

Table 6.4: Experiment 1a - p-values for the one-tailed paired t-test.

The p-values we received are presented in Figures 6.4 and 6.5. From the p-values for $\forall \epsilon \in \{0.1, ..., 0.9\}$ we can conclude that there is a statistical significance of the average of the methods we examined.

A visual representation of the aggregated data is presented in Figures 6.8 to 6.10. In particular, in these Figures we display the average revenue per epoch for each epoch and for each of the 100 runs for $\forall \epsilon \in \{0.1, 0.3, 0.5, 0.7\}$. For example, in Figure 6.8(a) we depict the points (x,y) where x is the average revenue of an epoch i, where $1 \leq i \leq 200$, during a run k, where $1 \leq i \leq 200$, when no trust model is used and y is the corresponding average revenue when the promise-based model is used. The light line we can observe in the diagrams in Figure 6.9 is $y = x$.

A data point (x', y')

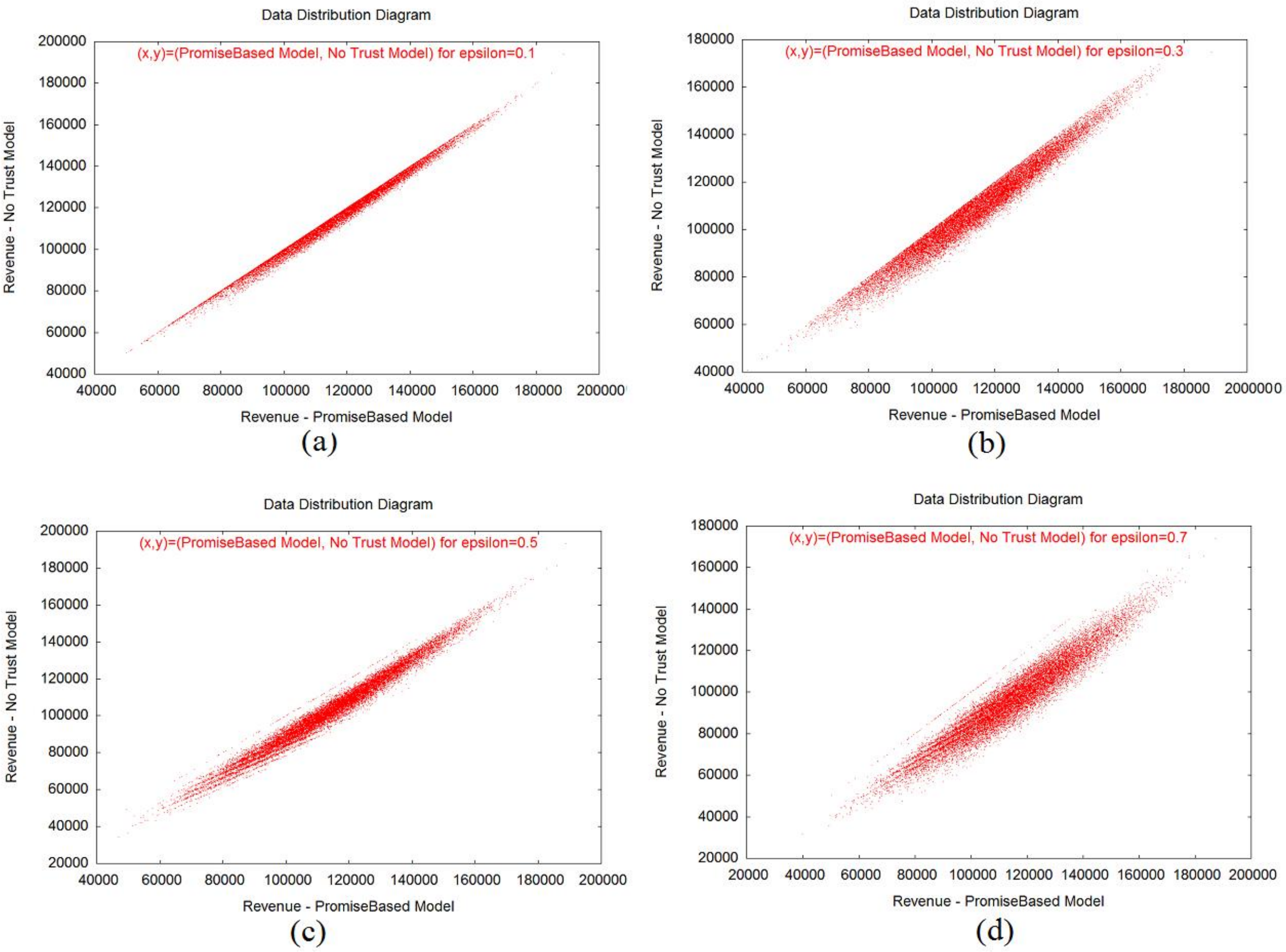
- on the left of the line $y = x$ means that $y' > x'$

ϵ	No Trust vs Promise-based	Promise-based vs Probabilistic	Probabilistic vs Maximum
0.1	0	1.4105E-136	0
0.2	0	2.34714E-84	0
0.3	0	8.80307E-76	0
0.4	0	1.8373E-141	0
0.5	0	5.65046E-47	0
0.6	0	8.10976E-20	1.5726E-205
0.7	0	9.99271E-12	4.6438E-131
0.8	0	1.20936E-07	2.97619E-70
0.9	0	1.7019E-05	6.15002E-41

Table 6.5: Experiment 1a - p-values for two tailed paired t-test.

- on the right of the line $y = x$ means that $x' > y'$, and
- on the line $y = x$ that $x' = y'$.

As we can see from the Figure 6.8 the promise-based model overall performs better than the case where no the trust model is used. Similar, in Figure 6.9 the probabilistic model overall performs almost equal to the promise-based model, and thus it also performs better than the case where no trust model is applied.



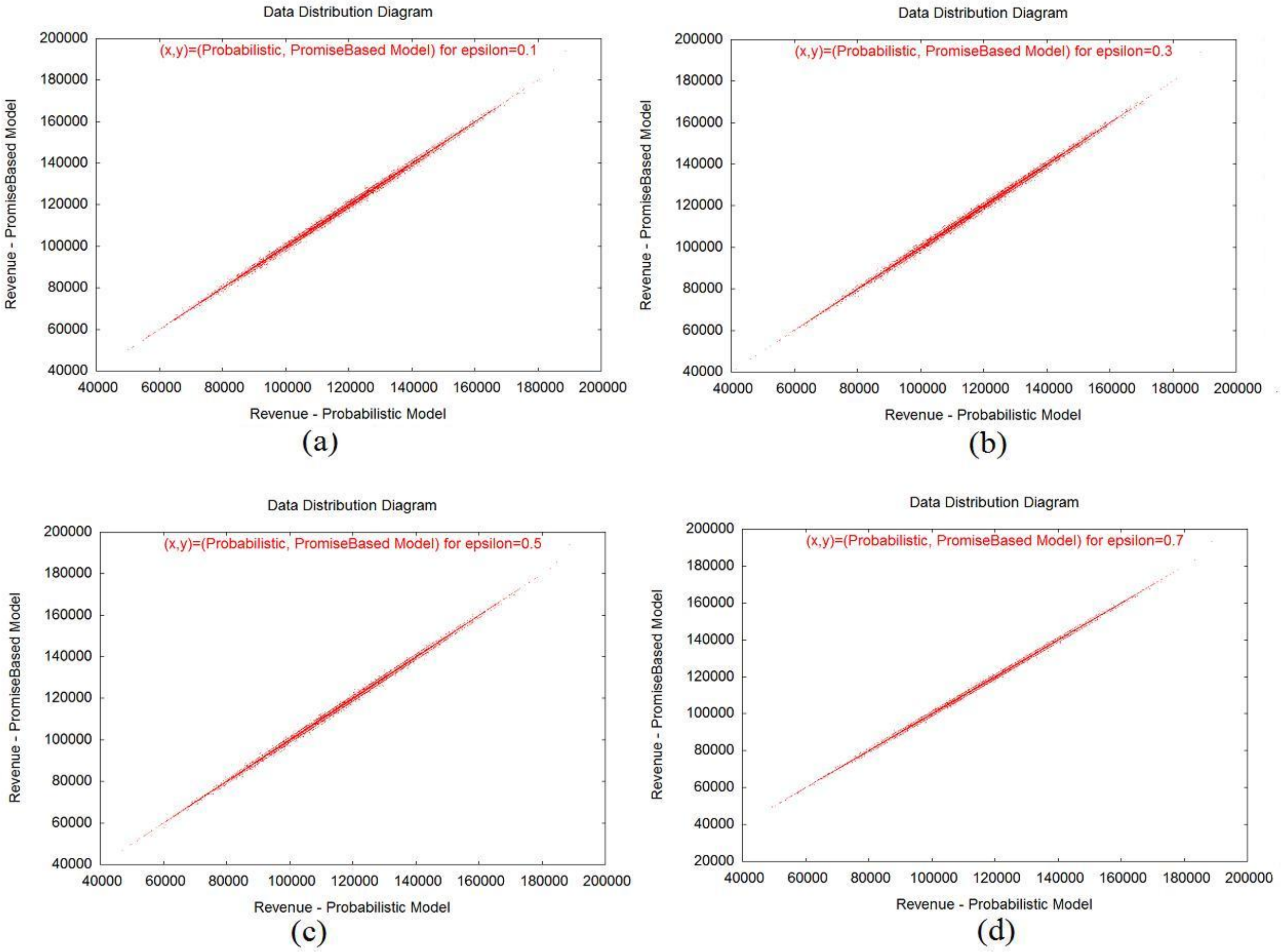
(a)

(b)

(c)

(d)

Figure 6.8: Experiment 1a. Data Distribution Diagram - No Trust vs Promise-based.



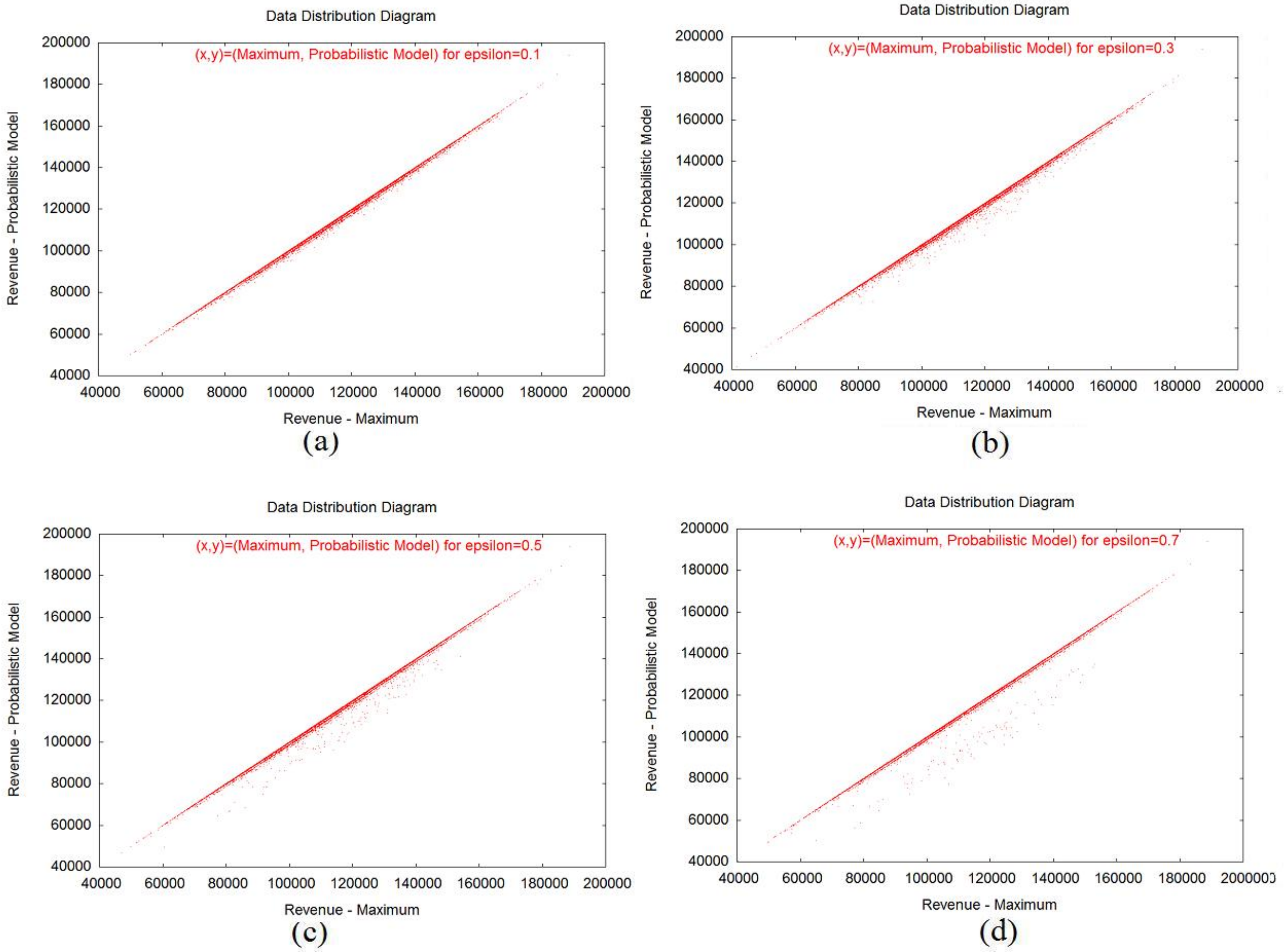
(a)

(b)

(c)

(d)

Figure 6.9: Experiment 1a. Data Distribution Diagram - Probabilistic vs Promise-based.



(a)

(b)

(c)

(d)

Figure 6.10: Experiment 1a. Data Distribution Diagram - Average Maximum Revenue vs Probabilistic.

Experiment 1b

In Experiment 1b we consider settings where the untruthful agents do not have information about the trustworthiness of other agents. This experiment essentially demonstrates the impact of Trust Property 4. Recall, Trust Property 4 says that *assume that agent a made promise $pm(s,a)$. Then for any $\epsilon > 0$ and for any $p' < pm(s,a)$*

$$Tr(pm(s,a), pm(s,a) + \epsilon) > Tr(p', p' + \epsilon)$$

and

$$Tr(pm(s,a), pm(s,a) - \epsilon) < Tr(p', p' - \epsilon)$$

Essentially, Experiment 1b is similar to the Experiment 1a with the following *difference*:

- The untruthful agents (that is the agents in A_{UT}) under-declare their contribution **only** when they have participated in the list L but not in S more than their "tolerance".

We considered exactly the same setting and values for the parameters as in Experiment 1a. Recall, the budget of the group G is selected in $[0, 8]$ through the uniform distribution, $|A_{UT}|$ is set to 500, the horizon to 10 which is the 2% of the agents in A_{UT}, $|S|$ to 4, γ to 0.5, $|L|$ to 50 which is the 10% of the agents in A_{UT} and the $Tolerance$ to 40 which corresponds to 2 epochs assuming that an agent was selected in L, but not in S, in all the iterations of 2 epochs. In each case we assumed that each agent in A_{UT} under-declares by the same amount ϵ, where $\epsilon \in \{i \cdot 0.1 | i = 1, ..., 9\}$. However, we would like to note that this is also a representative sample of numerous tests we performed.

In this experiment, we computed i) average maximum cumulative revenue per epoch the group G would have enjoyed if no agent was untruthful, ii) the average cumulative revenue of the group G per epoch for each of the following cases: no trust model is applied, the promise-based is applied, and the probabilistic model is applied, iii) the percentage of the average cumulative revenue with respect to the average maximum cumulative revenue of the group G per epoch for each of the latter cases.

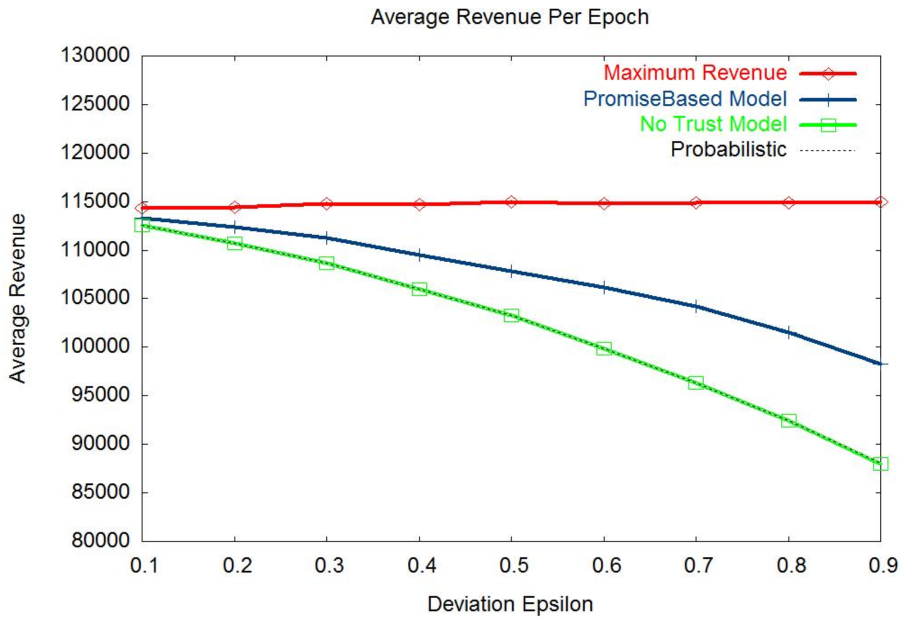


Figure 6.11: Experiment 1b - Average revenue per epoch.

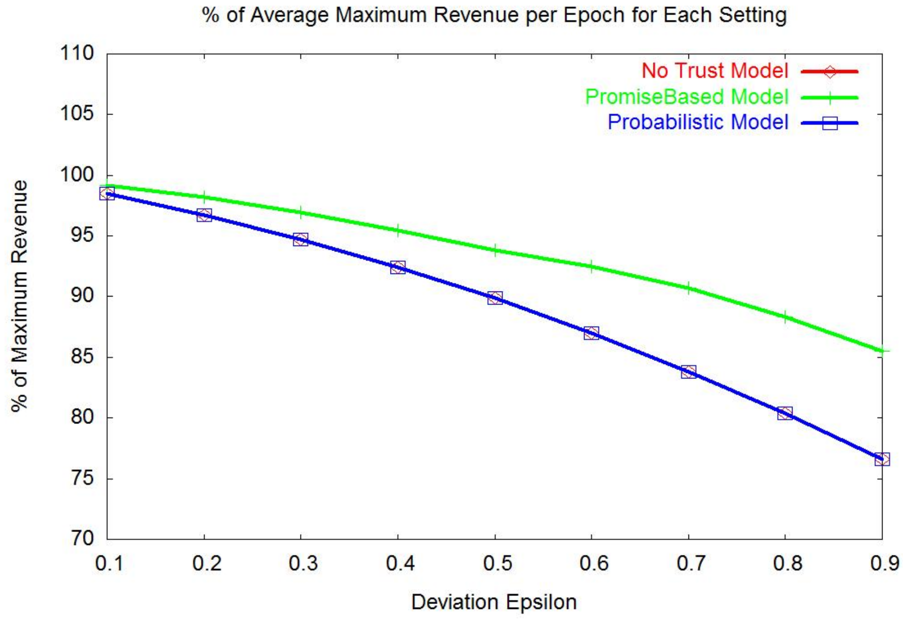


Figure 6.12: Experiment 1b - Percentage of the average maximum revenue per epoch.

ϵ	No Trust Model	PromiseBased Model	Probabilistic Model
0.1	98.47 %	99.148 %	98.471 %
0.2	96.696 %	98.175 %	96.698 %
0.3	94.713 %	96.967 %	94.716 %
0.4	92.431 %	95.476 %	92.435 %
0.5	89.894 %	93.842 %	89.902 %
0.6	87.005 %	92.489 %	87.012 %
0.7	83.82 %	90.695 %	83.829 %
0.8	80.402 %	88.361 %	80.415 %
0.9	76.572 %	85.476 %	76.583 %

Table 6.6: Chapter 6-Experiment 1b: Percentage of the maximum average revenue per epoch each model achieves.

ϵ	p-value
0.1	3.597E-26
0.2	3.9957E-127
0.3	0
0.4	0
0.5	0
0.6	0
0.7	0
0.8	0
0.9	0

Table 6.7: Experiment 1b - p-values for ANOVA Single Value Test.

Results - Experiment 1b.

We repeated each experiment 100 times and we present the average results in Figures 6.11 and 6.12. In particular, Figure 6.11 depicts the average revenue a group accumulates per epoch for the case where no trust model is used, the case where the promise-based model is used, the case where the probabilistic model, and the average maximum revenue which represents the maximum average revenue the group can achieve. As we can see, the promise-based trust model outperforms the probabilistic which essentially converges to the case where no trust model is applied. Figure 6.12 presents the percentage of the maximum revenue each model achieves. Similar to Figure 6.11 we can see that the promise-based model performs better than both the probabilistic model and the case where no trust model is applied. Also, we can see that the difference between the percentage the promise-based model achieves and the percentage the probabilistic and the no trust model case achieve increases as the deviation ϵ increases.

Statistical Analysis - Experiment 1b.

For this experiment we validated the significance of the difference between the average revenue for the case where no trust model was used, the case the promise-based was used, the case of the probabilistic is used, and the average maximum revenue. First we executed the *ANOVA single value test* [45] for the average revenue per epoch that was computed for each of the 200 epochs for each of the 100 runs. The results are presented in Table 6.7. Since for every ϵ the $p - value$ is less than 0.05 there is a statistical significance of their average values.

The next step was also to perform paired sample t-tests [82] for the following cases

- No Trust vs Probabilistic
- Promise-based vs Probabilistic
- Maximum vs Promise-based

ϵ	No Trust vs Promise-based	Promise-based vs Probabilistic	Maximum vs Promise-based
0.1	0.002392065	0	0
0.2	1.94E-08	0	0
0.3	8.33E-13	0	0
0.4	1.12E-16	0	0
0.5	4.40E-28	0	0
0.6	1.13E-24	0	0
0.7	5.32E-26	0	0
0.8	2.88934E-38	0	0
0.9	6.64E-32	0	0

Table 6.8: Experiment 1b -p-values for the one tailed paired t-test.

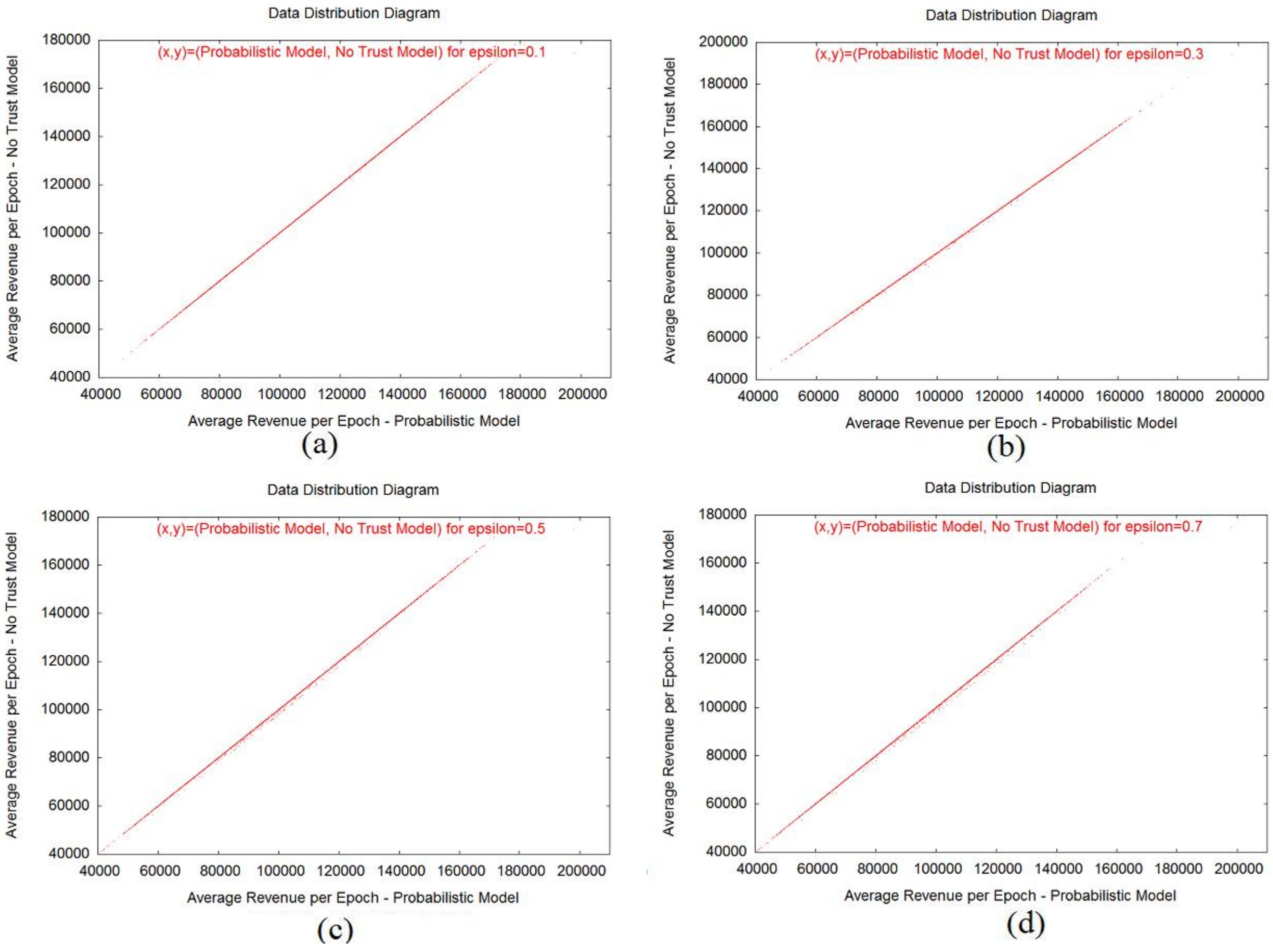
ϵ	No Trust vs Promise-based	Promise-based vs Probabilistic	Maximum vs Promise-based
0.1	0.004784129	0	0
0.2	3.88E-08	0	0
0.3	1.66685E-12	0	0
0.4	2.23E-16	0	0
0.5	8.79E-28	0	0
0.6	2.2514E-24	0	0
0.7	1.06E-25	0	0
0.8	5.77867E-38	0	0
0.9	1.32834E-31	0	0

Table 6.9: Experiment 1b - p-values for two tailed paired t-test.

The p-values we received are presented in Tables 6.8 and 6.9. From the p-values for $\forall\epsilon \in \{0.2, ..., 0.9\}$ we can conclude that there is significant difference between the methods we examine. For $\epsilon = 0.1$ there is also significant difference, besides the comparison of the case where no trust model is applied and the case the probabilistic model is applied. This means the probabilistic case converges to the no-trust case.

Similar to Experiment 1a, we present a representative sample of the visualization of the data. In particular, Figures 6.13 to 6.15 display the revenue per epoch for each epoch of a run and for each of the 100 runs for $\epsilon \in \{0.1, 0.3, 0.5, 0.7\}$. We would like to note that the straight line that we can notice in the above Figures falls on the curve $y = x$.

As we can see from the Figure 6.13 the probabilistic model converges to the case where no trust model was used. Given now that from Figure 6.14 it is obvious that the promise-based model performs better than the case where no trust model was used we can conclude that the promise-based performs better than both the probabilistic model and the case where no trust model was used.



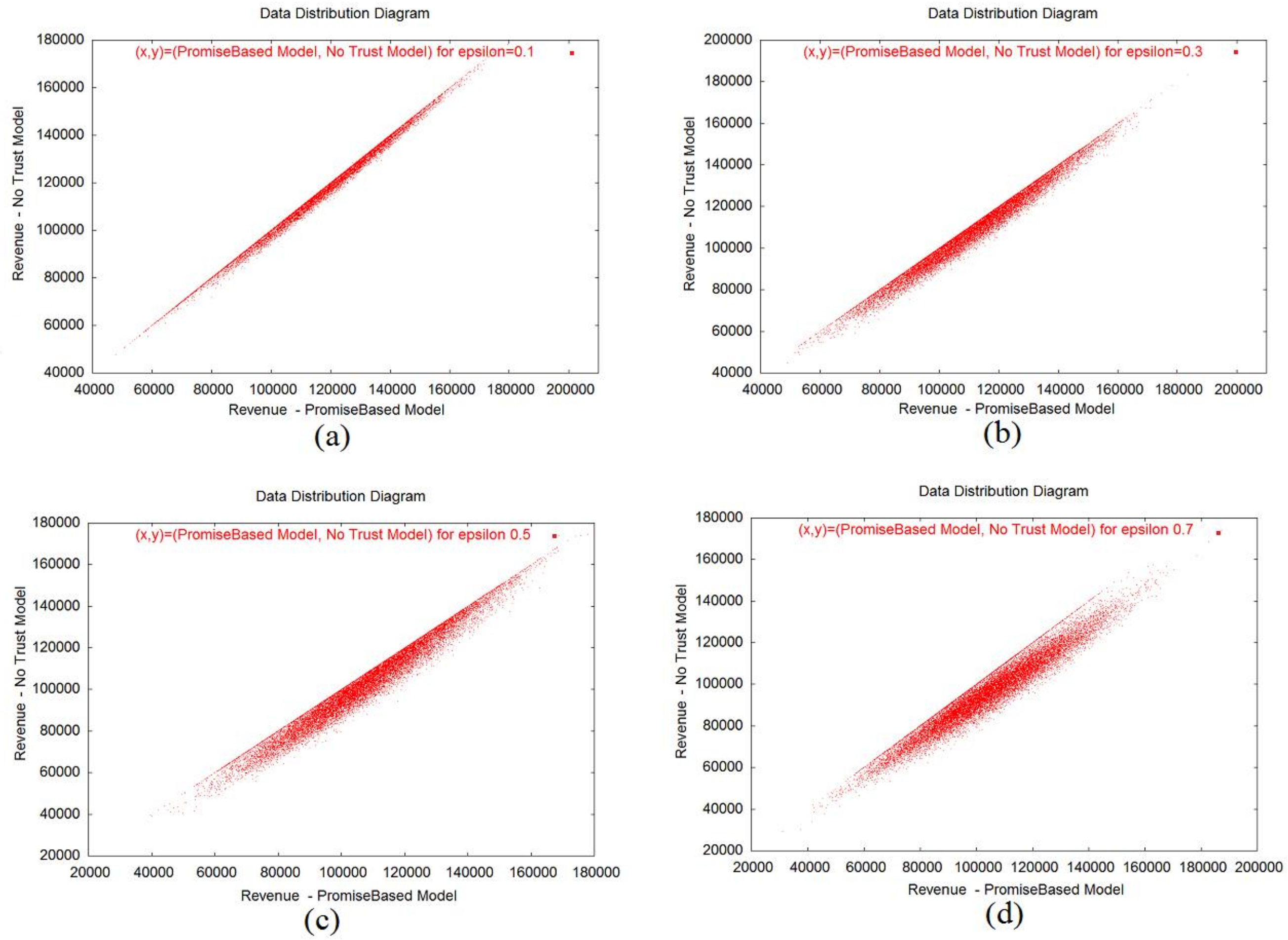
(a)

(b)

(c)

(d)

Figure 6.13: Experiment 1b. Data Distribution Diagrams - Probabilistic vs Probabilistic-based



(a)

(b)

(c)

(d)

Figure 6.14: Experiment 1b. Data Distribution Diagrams - No Trust vs Promise-based.

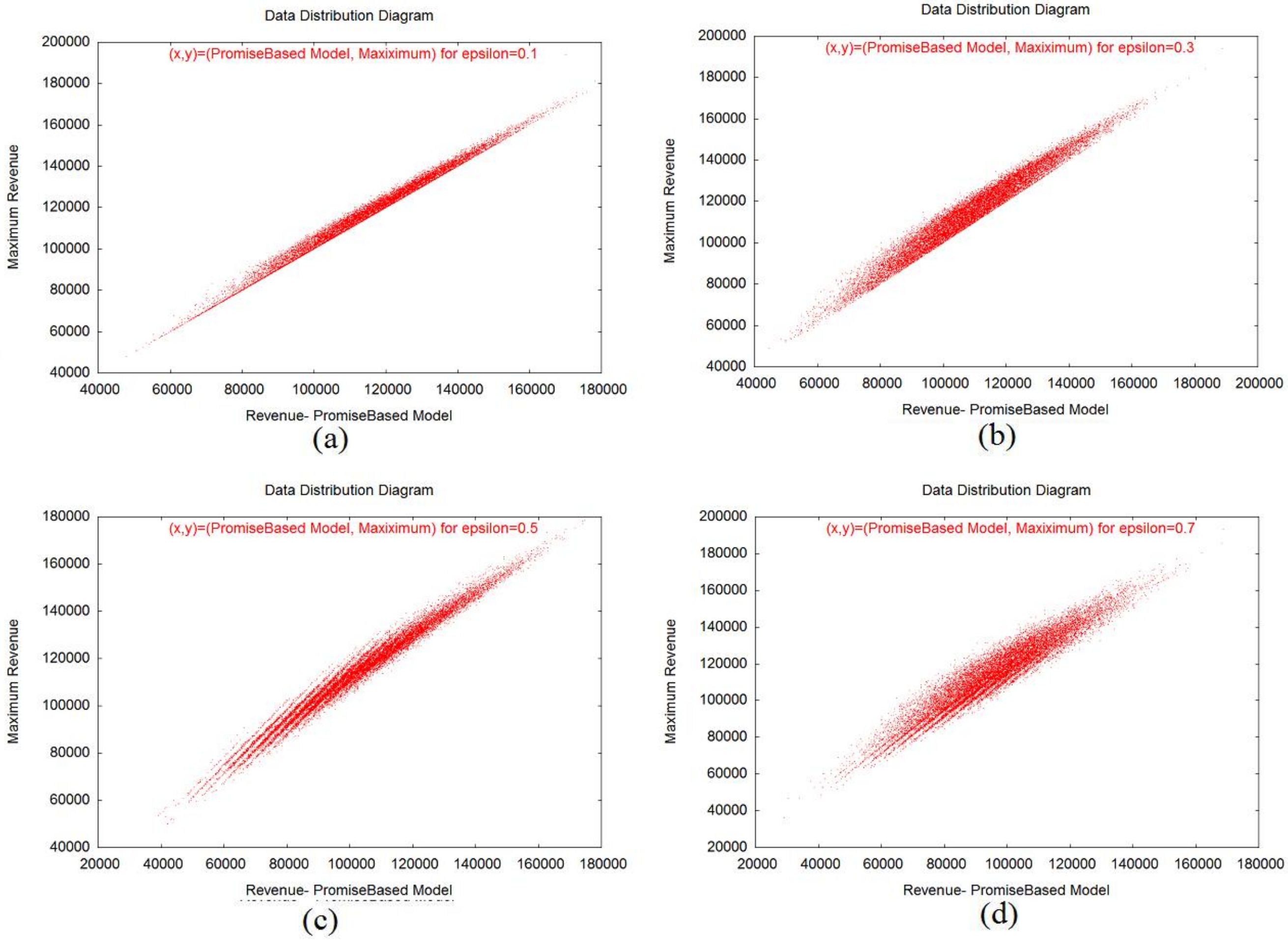


Figure 6.15: Experiment 1b. Data Distribution Diagrams - Promise-based vs Maximum.

Discussion - Experiment 1. Overall, experiment 1 shows that a simple deployment of a trust model can significantly improve the revenue of the truthful group. In other words, the experiment demonstrates that it is important to tackle the problem of over-delivering and that the idea of "the more the better" is not always desirable.

From *Experiment 1a* our general observation is that the probabilistic model appears to do marginally better. We believe the reason for this is that the instantiation of the promise-based trust function we used appears to be very lenient. It was very interesting to observe that even such a lenient instantiation of our proposed trust model can compete with the probabilistic model. The more lenient a trust model is the more space it can give for untruthful behaviour since by using it, it will take longer for deceptive behaviour to get reflected on the overall trustworthiness.

Experiment 1b considers settings where the untruthful agents do not have information about the trustworthiness of other agents and demonstrates the positive impact of Trust Property 4. As we can see, our promise-based model outperforms both the probabilistic model and the case where no trust model was applied. The reason we believe this happens is because it penalizes agents that promise less since this decreases the probability the group G can access the best possible plan. The higher the penalty in its trustworthiness an agent receives the more likely it is to not get selected.

The main advantage of our promise-based function in contrast to the probabilistic model is that it can be applied in settings where agents might not be able to provide exact promises. There are two main reasons why the probabilistic model is not suitable for these settings: i) for each deviation ϵ, no matter its value or in which range it occurs, the probabilistic model assumes it has equal significance allowing, in this way, the strategic agents to manipulate it (i.e., lie significantly and then be truthful in less important situations in order to boost up their trustworthiness), and ii) the probabilistic model either assumes that the truthful agents can always make an accurate promise or if a truthful agent deviates even by a little due to uncontrollable situations it is treated equally with an agent that deviates by a higher deviation. As we showed in section 6.2.3 our proposed properties take into consideration both the deviation for a promise and a delivered contribution, in addition to the range it occurs, while it incentivizes the agents to avoid strategic behaviour. Furthermore, the fact that the probabilistic model treats equally a deviation irrespectively whether this is the result of over-delivery or under-delivery results in Theorem 2 not being satisfied. As we already discussed, Theorem 2 is important since in settings where there is high uncertainty and truthful agents may unintentionally misreport their promises, risk-seeking agent can hide behind these truthful agents and play strategically. Theorem 1 essentially provides strong incentives to the risk-seeking agents to limit their strategic behaviour.

Concluding, we showed that i) the deployment of a trust model for settings where agents over-deliver their anticipated contribution can limit the strategic behaviour of untruthful agents and increase significantly the revenue of truthful agents, and ii) that our proposed model converges to the probabilistic model in a base-line scenario (experiment 1a) while it performs clearly better in settings where the untruthful agents do not have information about the behaviour of other untruthful agents (experiment 1b).

Since, to the best of our knowledge, there is no trust model that was designed to consider simultaneously both under-delivery and over-delivery, the deviation between two promises and the range it occurs, we cannot have a fair comparison that can demonstrate empirically the value of our proposed approach with respect to other proposed models. For this reason in the next experiment we try to examine if it would be possible to extend the probabilistic model in order to deal with both over-delivery and under-delivery.

6.4.2 Experiment 2

In settings where each agent is requested to perform a task or to provide information and there are only two possible scenarios (i.e., positive or negative completion) the most dominant method in the literature to reason about the trustworthiness of the agent is to use a probabilistic model [25, 76]. In particular, the trustworthiness of the agent is measured as the probability that a positive interaction will occur given the number of positive and negative interactions in the past. More specifically, the trustworthiness is measured as

$$trust = Prob(pos, neg) = \frac{pos}{pos + neg} \tag{6.21}$$

where *pos* is the number of positive interactions and *neg* is the number of negative interactions.[4]

Goal of Experiment 2: In settings where an interaction of an agent is characterized by two continuous variables, for example, the promise p the agent gave and the delivery d it end up providing, where $p, d \in \mathbb{R}$, in order to use the models in [25, 76] we need to decide based on the values of p and d, whether the interaction is successful or not. In cases where p represents a promised contribution and d a delivered contribution, an obvious way to do the conversion is to classify the interaction based on the percentage of the promised contribution p the delivered contribution d is. However, the ratio of d over p is not enough. If we would like, similar to our proposed model, to additionally consider the deviation of p and d and the range it occurred we need to also take into consideration the value of p.

Overall, the goal of this experiment is to examine if there is any notable relation between the percentage threshold for classifying a tuple (p, d) as a positive or a negative interaction, the value of the promised contribution p and the deviation $\epsilon = (p - d)$ such that the

[4] Note that Laplace smoothing can be done to avoid cases where $pos + neg = 0$.

assignment of a trust rating based on the probability $Prob(pos)$ follows the four trust properties we defined in section 6.2.3. We would like to do this in order to examine if there is a way to extend the probabilistic model to also consider settings like the ones our promise-based model was designed for.

Recall, the probability $Prob(pos)$ considers a set of interactions S in which every interaction is either characterized as positive or negative. For this reason, in our experiment we compare the $Prob(pos)$ over a set of interactions S with the average trust score Tr per interaction in S. We will say that $Prob(pos)$ follows our four trust properties if each time achieves a value equal to Tr. Our goal is to find how to translate each interaction in S such that $Prob(pos) = average(Tr)$.

Methodology: We will use our *i-Functions* as a guide to decide to which binary rating (i.e., positive or negative) a tuple (p, d) should convert. More specifically, our ultimate goal is to define a function $\omega(\epsilon, p)$ such as for $\forall \epsilon \in [0, 1]$ and $\forall p \in [0, 1]$ it will provide the percentage of the acceptable delivered contribution with respect to the promised contribution that should result in a positive rating. In particular:

- **Under-delivery**: If the delivered contribution d is at least equal to the $\omega\%$ of the promised contribution p (i.e., $100 \cdot d \geq p \cdot \omega$) the transaction is characterized as positive and as negative otherwise.
- **Over-delivery**: If the delivered contribution d is at most equal to the $\omega\%$ of the promised contribution p (i.e., $100 \cdot d \leq p \cdot \omega$) the transaction is characterized as positive and as negative otherwise.

Setting: For our experiment we considered contributions in the interval $[0, 1]$. We divided this interval to nine sub-intervals $\{[0.1, 0.2], [0.2, 0.3], ..., [0.9, 1.0]\}$. For each of these intervals we find the values for $\omega(\epsilon, p)$ for every deviation $\epsilon = p - d \in \{0.2 \cdot i | i = -24, ..., 24\}$. In each iteration we generated same number (=$AggregationNum$) of samples of tuples (p, d) such that $p \in [a, b]$ and $d = p + \epsilon$ by using the uniform distribution. We then computed the average values $d' = average(d)$ and $p' = average(p)$ of d and p, respectively, and then the trust $Tr(p', d')$. The trust function $Tr(p, d)$ we consider is:

$$Tr(p, d) = (1 - |d - p|)(\frac{1}{2} + \frac{d^2 - p^2}{4})$$

The next goal was to find the value of the ω function that leads to a discretization of the tuples (p, d) to positive or negative, which results in a probability $Prob(pos) = \frac{pos}{pos+neg}$ equal to $Tr(p', d')$, where pos and neg is the number of positive and negative interactions, respectively.

The trust function Tr we used in our experiment is the one presented in equation 6.14 with $n = 2$. We repeated the latter procedure 1000 times and we calculated the average ω value for each $\epsilon \in \{0.2 \cdot i | i = -24, ..., 24\}$ and for each interval $\{[0.1, 0.2], [0.2, 0.3], ...,$ $[0.9, 1.0]\}$. Here, we would like to note that an important parameter is the total number of ratings (i.e., $pos + neg$) in order to calculate the probability $Prob(pos)$. We will refer to this number as the *aggregation number* and we will denote it by $AggregationNum$. We repeated our experiment for $AggregationNum = 10$, 30 and 100.

In particular, the algorithm works as follows:

Experiment 2

```
for all $\epsilon$ in $\{-0.48, -0.46, ..., 0.48\}$ do
  for all $[a, b] \in \{[0.1, 0.2], [0.2, 0.3], ..., [0.9, 0.1]\}$ do
    for all k from 1 to 1000 do
      for all j from 1 to $AggregationNum$ do
        Assign $p[j]$ a random value in $[a, b]$
        $d[j] \leftarrow p[j] + \epsilon$
      end for
      Find $p' = avg(p)$ and $d' = avg(d)$
      $TRUST = normalized(Tr(p', d'))$
      Find the value of the function $\omega$ such that if we convert the tuples $(p[], d[])$ to
      binary events the probability $Prob$ a positive event to occur has the minimum
      deviation from $TRUST$.
    end for
    Return average values for $\omega$, $d'/p'$, $Tr$, and $Prob$
  end for
end for
```

In order to convert a tuple (p,d) to a binary rating we used the following algorithm:

Convert tuple (p,d) to a binary rating
Input: Promised contribution p, Delivered contribution d, Percentage ω
Output: Type of binary rating

```
if $d < p$ then
    if $100 \cdot d/p < \omega$ then
        return -1
    else
        return 1
    end if
else
    if $(100 \cdot d/p) > (\omega)$ then
        return -1
    else
        return 1
    end if
end if
```

Essentially, for the case of under-delivery a tuple (p, d) results in a positive rating if $100 \cdot \frac{d}{p}$ is greater or equal to the percentage ω and to a negative rating otherwise. In the case of over-delivery a tuple results in a positive rating if $100 \cdot \frac{d}{p}$ is less or equal to the percentage ω.

Assumptions: The promised contributions that are aggregated lie in the same interval and they differ from the delivered contributions by ϵ.

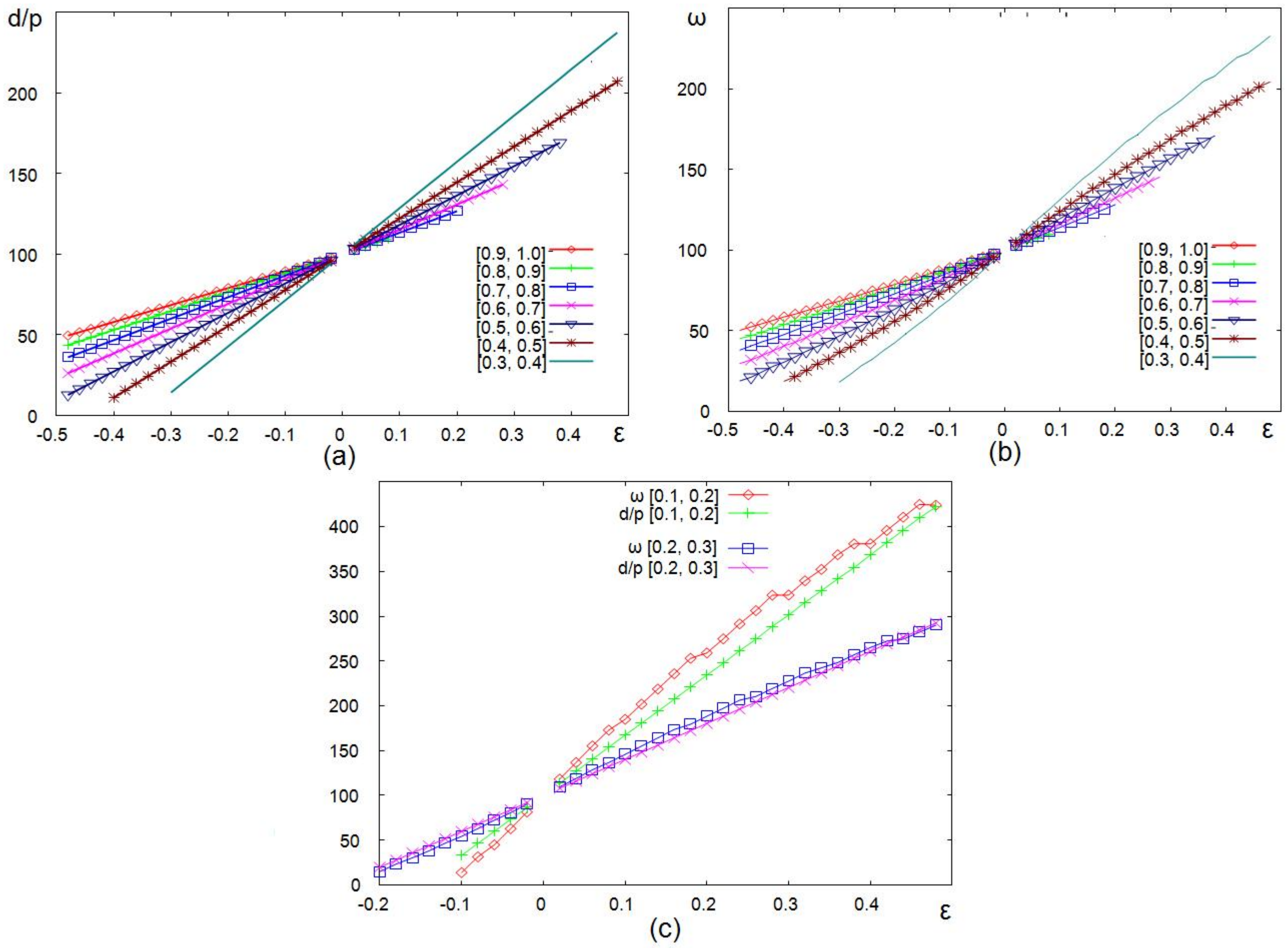


Figure 6.16: The diagrams of ω and $\frac{d}{p}$ for aggregation over 10 ratings for the intervals $\{[0.1, 0.2], ..., [0.9, 1]\}$.

Results & Conclusions: The results from our experiment are depicted in Figures 6.16 to 6.21.

In particular, in Figures 6.16 and 6.17 we present the value of ω and $\frac{d}{p}$ for every $\epsilon \in [-0.48, 0.48]$ and for each interval in $\{[0.1, 0.2], ..., [0.9, 1.0]\}$, where d is the average delivered contribution and p is the average promised contribution over the number of ratings we aggregate. Our first observation from Figures 6.16 and 6.17 is that the ω function appears to follow the same curve with the percentage of the average delivered contribution divided by the average promised contribution (i.e., $\frac{d}{p}$). If the curves of ω and $\frac{d}{p}$ overlapped that would meant that "given a set S of tuples (p, d) in order to convert each tuple to a binary rating in a way that the probability of a positive rating to occur to be equal with

$$AVG(Tr) = \frac{\sum_{\forall (p,d) \in S} Tr(p, d)}{|S|} \tag{6.22}$$

we just have to consider whether the value of $\frac{d}{p}$ for each rating is greater or less than $\frac{average(d)}{average(p)}$ over the set S."

However, if we compute the difference $\omega - \frac{d}{p}$ for each interval and for each value of ϵ we can observe that ω and $\frac{d}{p}$ deviate. This is presented in Figures 6.18 and 6.19. An interesting observation is that the difference $\frac{d}{p} - \omega$ (Figure 6.18) appears to follow the same type of curves as the trust function (Figure 6.1). As we can see in Figure 6.1 the Tr function for $[0, p)$ is convex and for $[p, d]$ is concave. Furthermore, the rate Tr first increases in $[0, p]$ is higher than the rate Tr decreases in $[p, 1]$. Similar, is the observation about the ω. For example, as we can see in Figure 6.19 for ϵ in $[-0.48, 0]$ the ω curve for any of the intervals is convex while for ϵ in $[0, 0.48]$ is ω curve concave. We believe this behaviour of the curve ensures that over-delivering by a deviation $\epsilon' > 0$ should result in a higher trust value (and thus more positive ratings) than under-delivering by the same deviation (i.e., $-\epsilon'$).

At this point we would like to note that the aggregation number, $AggregationNum$, influences the precision of the ω curve. For example, for $AggregationNum = 10$ we can only get a probability value in the $\{0.1, 0.2, ..., 1\}$ the more we increase it the better we could approximate the value Tr. This is depicted clearly in Figures 6.22 and 6.23. Note, that the ω, $\frac{d}{p}$, $\omega - \frac{d}{p}$, and Tr & $Prob$ diagrams for $AggregationNum$ equal to 100 are also depicted in the Appendix.

Use of the results: Given the data points of the $\omega(\epsilon, p)$ function that our experiment produced for each interval and for each $\epsilon \in [-0.48, 0.48]$ we might be able to find a good approximation of the $\omega(\epsilon, p)$ function for every $\epsilon \in [-0.48, 0.48]$. For example, for $\epsilon = 0.45$ for the interval $[0.6, 0.7]$ we know that $\omega(-0.24, p)$ is 62.95 while $\omega(-0.26, p)$ is 60.07. We can consider that $\omega(-0.265, p)$ lies on the straight line that connects the points

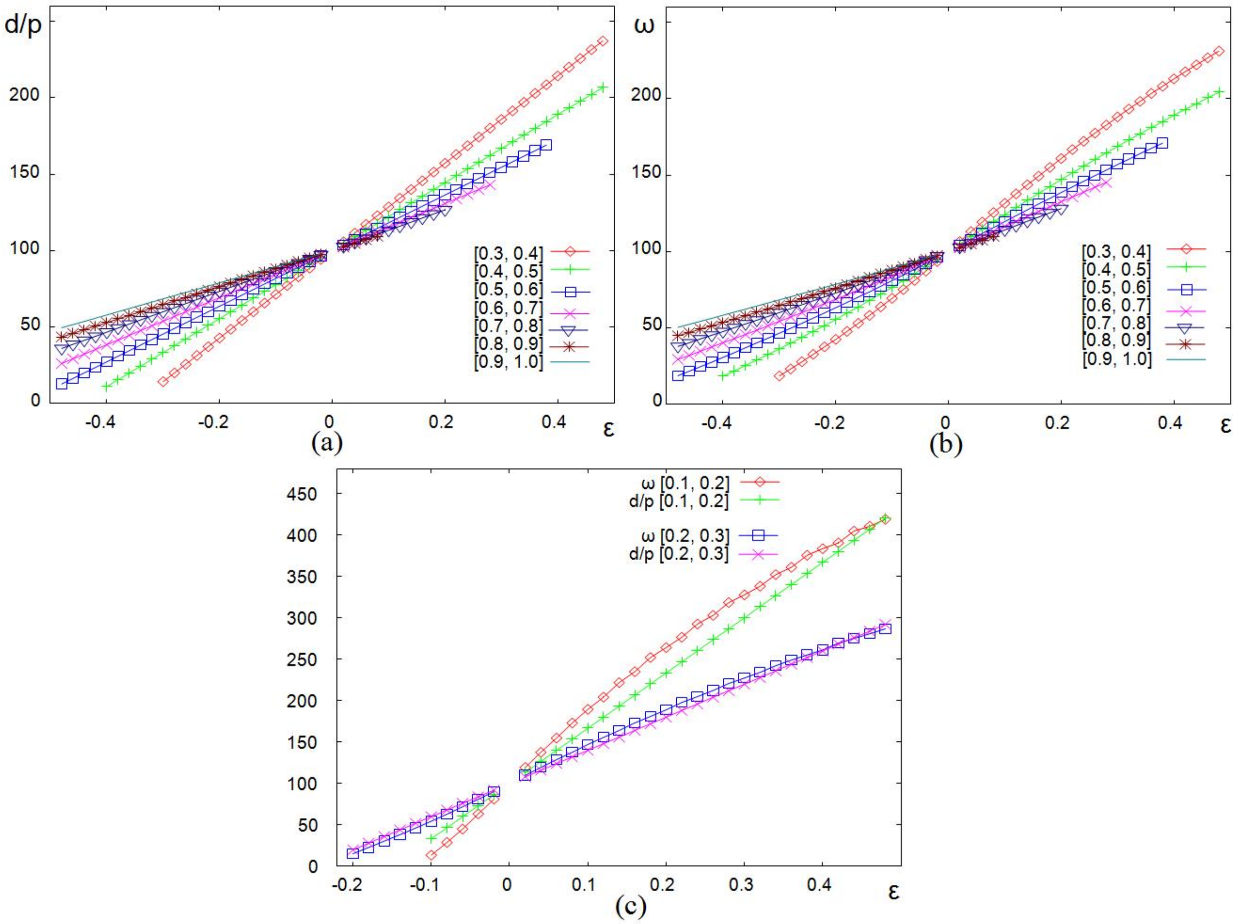


Figure 6.17: The diagrams of ω and $\frac{d}{p}$ for aggregation over 30 ratings for the intervals $\{[0.1, 0.2], ..., [0.9, 1]\}$.

$(-0.24, 62.95)$ and $(-0.26, 60.07)$ and thus calculate its value. Although the curves of ω in Figures 6.18(a)and 6.19(a) are for $p = \{0.15, 0.25, ..., 0.95\}$ given the pattern they follow, we can approximate the curve for any $p \in [0, 1]$.[5] Furthermore, an observation that might allow us to define more accurately the function ω is that the local minimum of $\omega - \frac{d}{p}$ for $\epsilon < 0$ slightly increases as the interval increase (i.e., going from $[0.4, 0.5]$ to $[0.5, 0.6]$).

In our current experiment we considered that the aggregated tuples (p, d) are such that every p lies in the same interval. However, more experiments are needed in order to determine how to aggregate tuples (p, d) such that the promised contributions p may belong in different intervals.

[5] By considering the average of $AggregationNum$ number of p given by the uniform distribution for

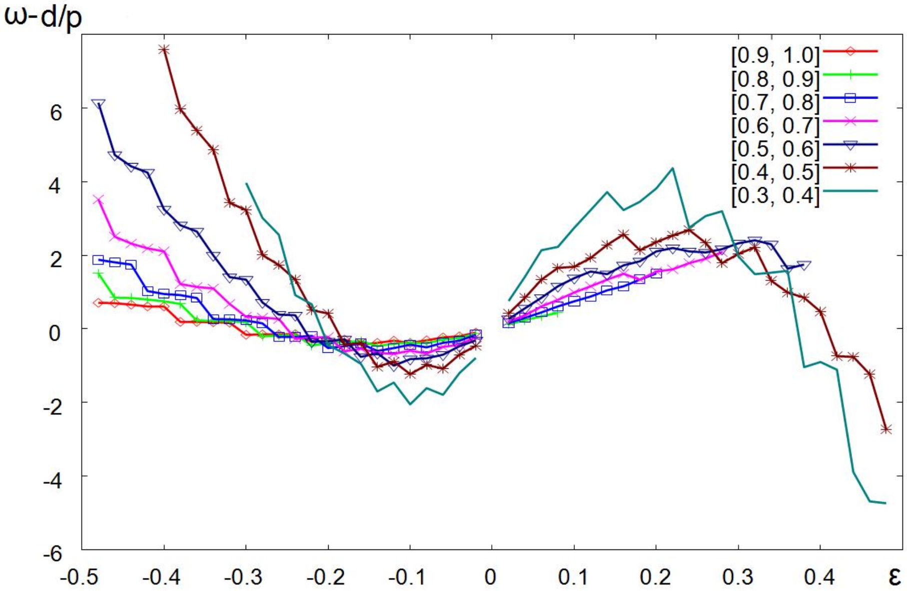


Figure 6.18: The diagrams of $\omega - \frac{d}{p}$ for aggregation number equal to 10 for each interval in $\{[0.3, 0.4], ..., [0.9, 1]\}$.

each interval $[\alpha, \beta]$ the average value of p converges to $\alpha + \frac{\beta+\alpha}{2}$.

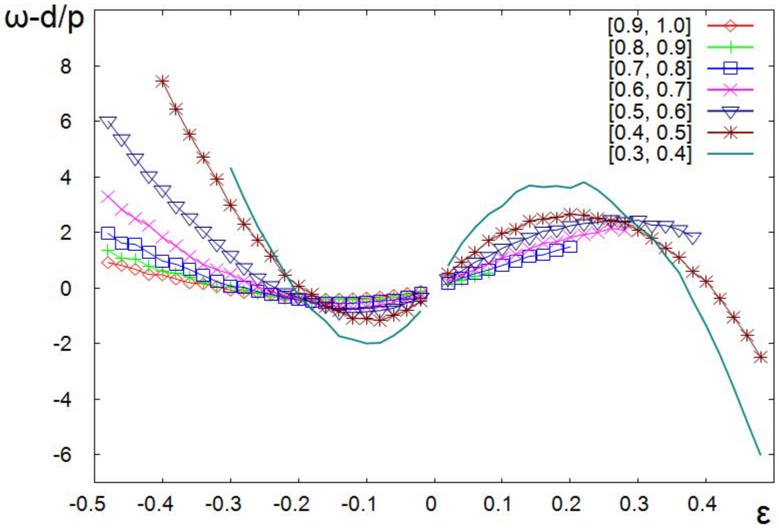


Figure 6.19: The diagrams of $\omega - \frac{d}{p}$ for aggregation number equal to 30 for each interval in $\{[0.3, 0.4], ..., [0.9, 1]\}$.

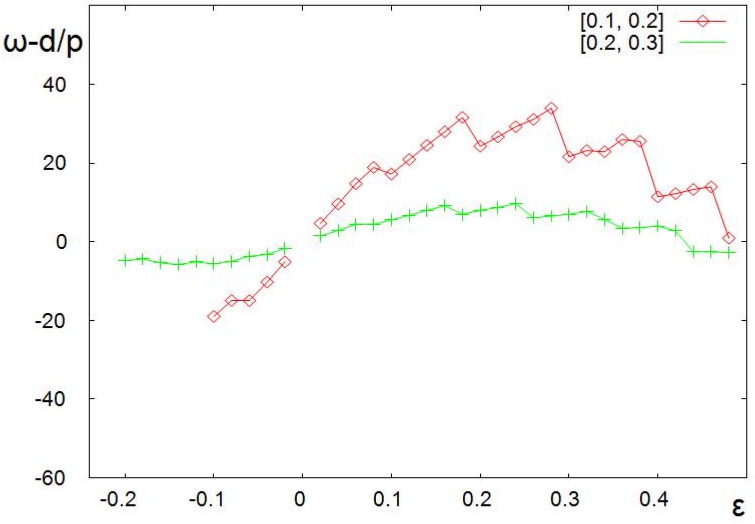


Figure 6.20: The diagrams of $\omega - \frac{d}{p}$ for aggregation number equal to 10 for each interval in $\{[0.1, 0.2], [0.2, 0.3]\}$.

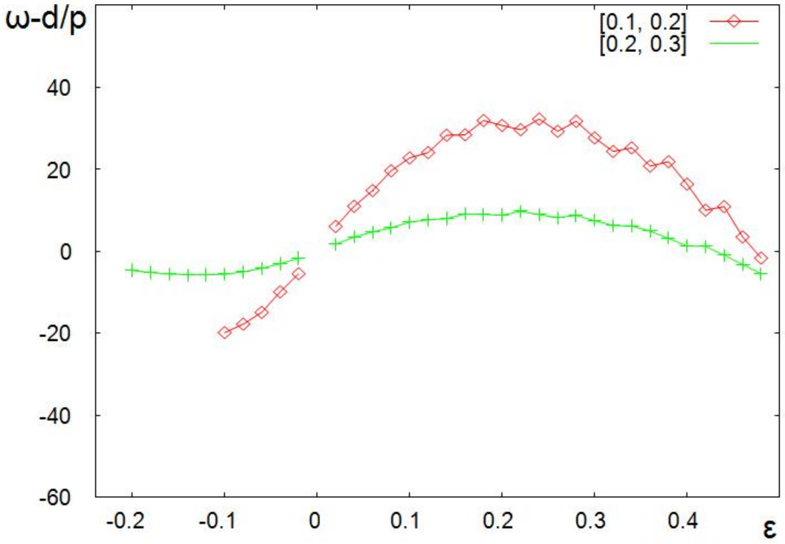


Figure 6.21: The diagrams of $\omega - \frac{d}{p}$ for aggregation number equal to 30 for each interval in $\{[0.1, 0.2], [0.2, 0.3]\}$.

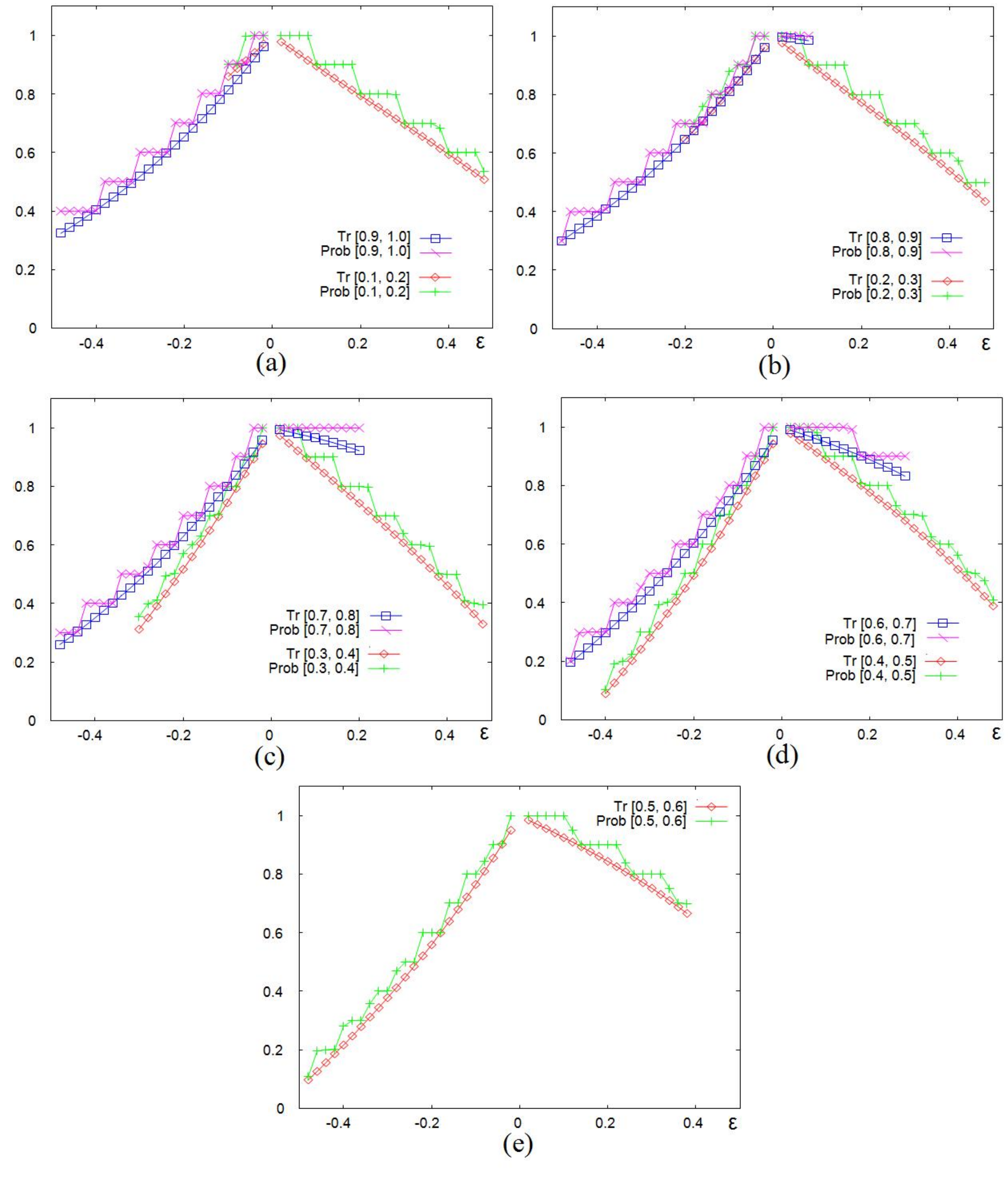


Figure 6.22: The average probability and average trust values for aggregation number equal to 10.

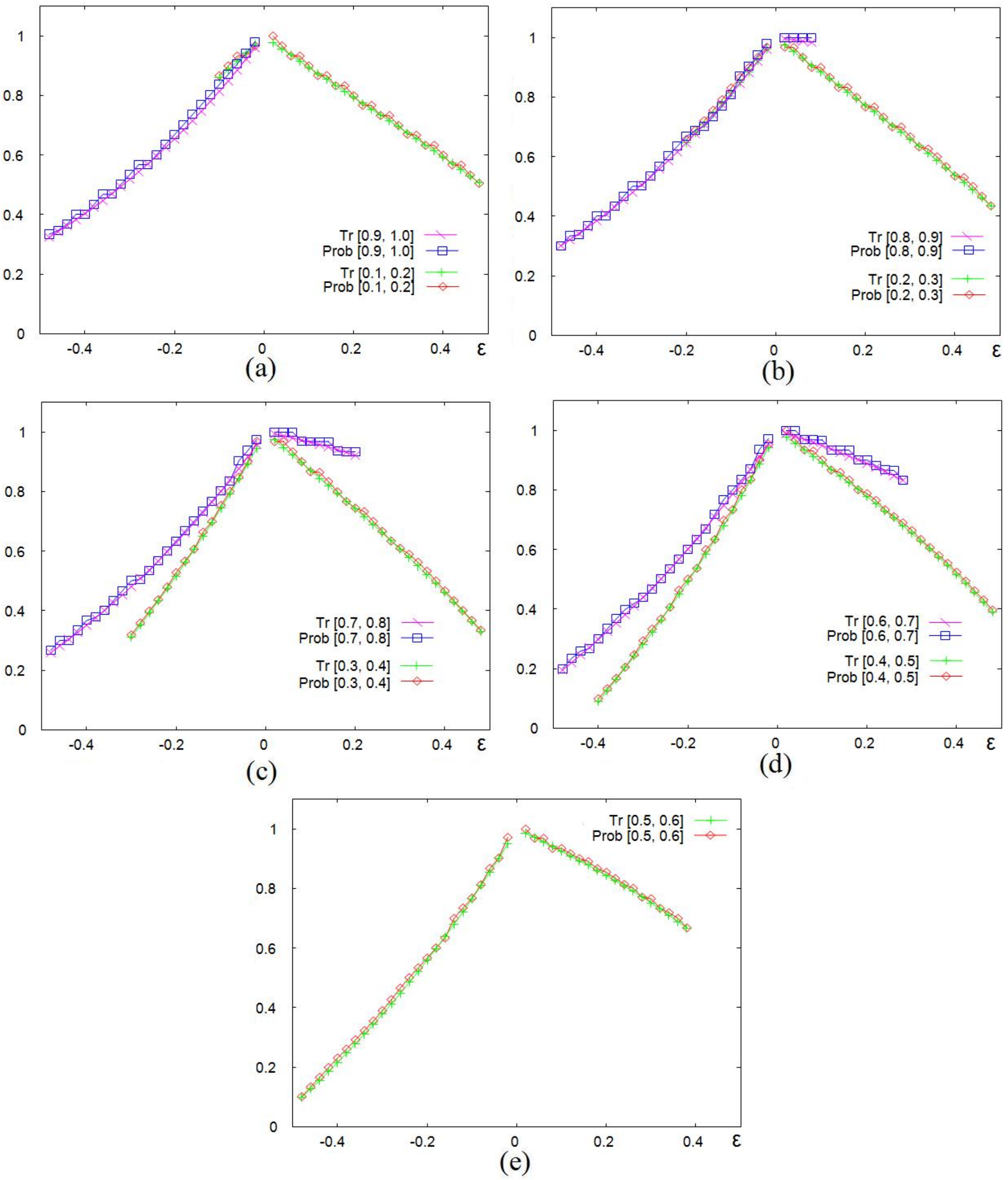


Figure 6.23: The average probability and average trust values for aggregation number equal to 30.

6.5 Summary

In this chapter, we proposed and formalized the minimum desirable properties a promise-based trust model should exhibit for settings in which important decisions are made based on the promised contributions of the participants. We showed that these properties ensure that an agent maximizes its trust-score by both delivering what it promised and by promising what it can deliver and we provided functions which embody these principles. Furthermore, we discussed how our model can be adapted to settings where, due to changing circumstances, an agent may be unsure when they make a promise that they will be able to deliver as desired. Our proposed model discourages agents from both over-stating and under-stating their potential contributions to the community. This is due to the fact that we are interested in designing a model that can be also applied in settings where the community might have to choose to follow a plan based on its current available resources, and thus it is crucial both to know the exact services it has, and to provide strong incentives to truthfully reveal the resources they can offer [4]. When the agents under-deliver on their promise, the community might engage in a plan of lower significance, since based on the reported resources the community has a false indication of its current resources. If the agents over-deliver on their promise, then this might result in agents deviating from the initial plan they committed to follow. Finally, our empirical evaluation showed that even a simple deployment of a trust model improves significantly the utility of truthful agents in settings where agents tend to under-declare their anticipated contributions. In settings where the untruthful agents have accurate information about other untruthful agents depending on the instantiation of the promise-based trust function we use, the probabilistic can marginally outperform our promised-based trust function while in settings where no such information is available our promise-based model clearly outperforms the probabilistic model.

Chapter 7

Discussion

In this thesis we focused on multiagent communities in which a significant portion of the available services is determined by the contributions of their participants. Examples of these communities include peer to peer systems, electronic markets, wikis, question/answer sites, and recommendation systems. Our goal is to:

- assist communities to be more effective in providing a beneficial environment that will be attractive to newcomers.

We achieve our goal by promoting the exchange of evaluation information between communities and by providing an explicit framework that enables the exchange to happen in a way that promotes honest reporting. The success of information exchange strongly depends on the quality of the exchanged data. Thus, we also proposed a model for communities to evaluate the trustworthiness of their agents. More specifically, we:

- provided a two phase mechanism that facilitates the truthful exchange of information between communities regarding their agents
  1. *First phase:* Advisor selection.[1]
  2. *Second phase*: Advisor payment distribution.
- presented an effective metric to depict the trustworthiness of an agent based on the promises it honors.

In the first phase we determined the set of advisors to ask and in the second phase we determined their compensation. More specifically, in the first phase our goal is to provide

[1] Recall, an advisor is a community that provides information about its agents.

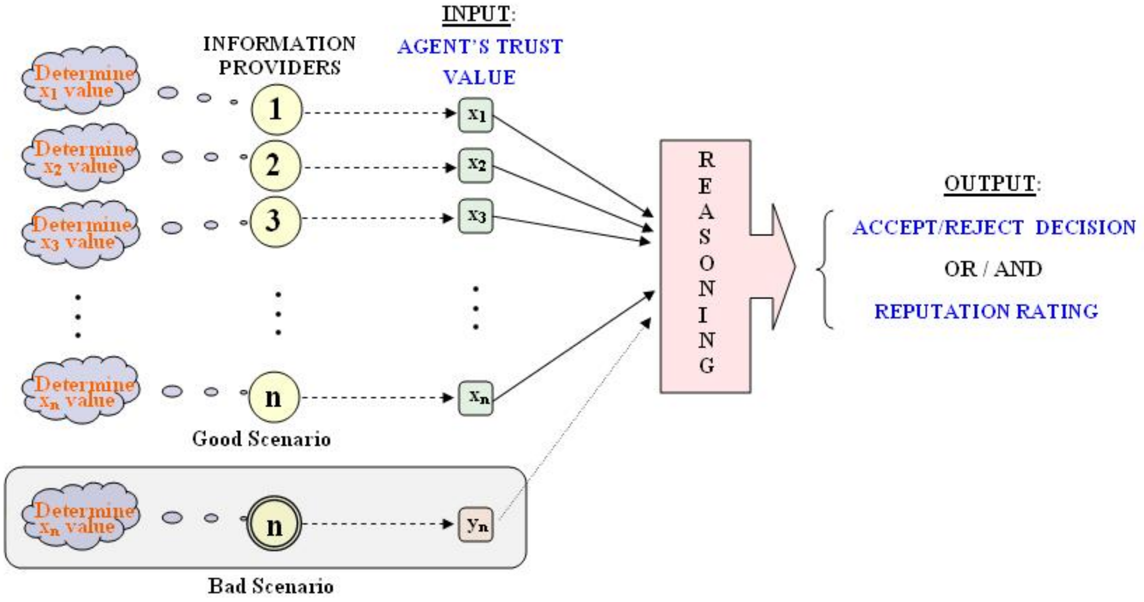


Figure 7.1: Reasoning Procedure

additional incentives for an advisor community to truthfully report the evaluation of all of its agents and in the second phase our goal is to compensate the advisors through a payment that considers the usefulness of the information they provided.

At a first glance the problem of exchanging evaluation ratings appears to be the same with the one of exchanging ratings between agents for settings where the agents are self-interested and have strong incentives to play strategically and occasionally lie. Since completely preventing strategic behaviour (i.e., pretend to be trustworthy in order to build reputation and then suddenly deceive other participants) is a challenging problem the extent it can be limited is important. This is the point where information exchange between communities slightly differs from information exchange between agents. In the case of communities, in order for a community to participate in the exchange it has to have guarantees that the other communities have strong incentives to be truthful in every step of the exchange procedure. In our setting, these steps include the selection procedure of the advisor, the distribution of the payments and the accurate evaluation of the agents.

In this thesis the biggest challenge with respect to deceptive behaviour ([9, 14, 38]) was to prevent communities from building good reputation by truthfully exchanging information about their mediocre agents and lying about their very best or their worst agents. For example, consider an e-market community C_A that until now is trustworthy and is about to provide information for one of its very best sellers to another e-market community C_B. The community C_A might choose to lie since it might be afraid that if it tells how wonderful the seller is, the community C_B will accept it. Having community C_B accept the agent might not be always desirable to C_A since the seller might have to share its limited merchandize between C_A and C_B. Moreover, things get more complicated since besides having communities play strategically and occasionally lie about certain types of agents, there might be agents that deceive communities and although communities are truthful about their agents, the agents change their behaviour. Determining whether the community lied or the agent deceived the community is a challenging problem. Communities can try to use the agents' strategic behaviour to their advantage. In particular, they can "hide" behind agents and pretend that it was not them who lied but the agent who was deceitful to them.

An implicit assumption that characterizes exchange mechanisms is that the available products are valuable to the recipient parties. In our setting the exchange product is the evaluation of the behaviour of agents, and thus it is crucial the evaluation mechanism, that each community implements, produces information that is valuable to the communities that purchase it. The evaluation of an agent can be measured either by its reputation or its trustworthiness. The reputation of an agent is a metric that depends mainly on the opinions of others and can be influenced significantly by the current situation of a community [29]. However, since trust is mainly measured based on direct experience [27, 57, 76] it can be more objective than reputation, and consequently more valuable. Due to this we suggest the exchanged information should be the trustworthiness of the agents and we present a trust model that can produce this type of data.

7.1 Advisor Selection and Payment Distribution

Advisor Selection

Restricting the set of advisors in a context where only the advisors consulted receive a compensation (i.e., payment) can be the basis of an incentive mechanism in which the advisors are motivated to behave in a desirable way. However, the way the set is determined is critical for the success of the mechanism. A first approach is to select n advisors that have the highest reputation or trustworthiness in providing advice. In our approach, we considered a mechanism that decides which advisors to consult when their evaluation lies in a range that does not clearly indicate which advisor is a better choice and thus the ordering of the advisors with respect to their evaluation rating does not provide sufficient information. For example, consider the existence of three advisors a, b and c with ratings 0.854, 0.850, and 0.848, respectively. Although the advisor a appears to be the best choice in reality c might be the best to ask. For example, a might have had a very good reputation or high trustworthiness but she might be inactive for quite a while, while c is now building his reputation and his rating is the result of recent interactions. In our work, we suggest a method that exploits the consistency among past advice and we show that if this selection takes place carefully, then strong incentives for advisors to be truthful are provided.

There is a number of different strategic behaviours the advisors can follow which essentially result from deploying inaccurate evaluation models [9, 14, 38]. However even if an accurate evaluation model is deployed, an advisor selection mechanism can be prone to strategic behaviour as exemplified by the following case. A community might observe that other communities provide truthful information with probability 0.8 due to model inaccuracy. If its agents are not malicious the community might translate the above probability into assuming it can afford to lie a bit less than 20% of the time and still be considered trustworthy. For example, consider five communities C_A, C_B, C_C, C_D and C_E. Assume that C_E has noticed from its own experience that C_A, C_B, C_C and C_D provide truthful information with probability 0.7, 0.75, 0.8 and 0.75, respectively. If C_E knows that the selection of the communities is based exclusively on the latter probabilities it can infer that it does not have to be always truthful. In particular, it can afford to lie at least 19% of the time. Thus, a new approach that will prevent a community from having sufficient information to efficiently decide when to lie or not is needed. Our key idea towards the solution of the problem is the exploitation of consistency among advice.

Our observation is that by exploiting the relation an information provider has with other information providers, which are known to be truthful, we can significantly limit strategic behaviour. In our approach, this is achieved through our selection algorithm, which considers the consistency of a community with other truthful communities. So, in simple words, a community must be consistent with as many truthful communities as possible which also tend to be consistent with each other. Thus, a community is better off by always being truthful.

Approaches like those proposed by Josang and Ismail [25], Teacy et al. [76] or Tran and Cohen [77] assume that the exchange of ratings is free and mainly focus on aggregating the exchanged rating and discarding or discounting the ratings that appear to be inaccurate. These models focus on the reasoning and the decision part of the mechanism depicted in Fig. 7.1 and essentially simply transfer the problem to the decision when to discard or discount the ratings. Since determining whether some ratings should be discarded (or discounted) is a very difficult problem we focused on incentivising the communities to truthfully report these ratings. In other words, our aim is to eliminate the “Bad Scenario” in Fig. 7.1 so the reasoning model has accurate data to process and does not significantly count on deciding which data to discount or discard. Other researchers have looked at the problems of providing incentives to models [31, 90]. However, their models appear to be prone to strategic behaviours like the ones demonstrated in [9, 14].

Although we are not the first to incorporate a graph-based approach in the design of incentive mechanisms, to the best of our knowledge our approach of exploiting the consistency of the nodes in a graph in order to promote honesty is novel. Our goal is to identify the strongest node in such a way that provides strong incentives to the communities (that are represented by the nodes) to truthfully report their information. With our approach it is harder for a community to play strategically (i.e., determine successfully when to lie and when not). It is important to note, however, that our focus is not only to determine the most truthful communities to ask based on past experience but also to provide them with strong incentives to continue being truthful in the future.

Hendrix et al. [22] propose an approach that exploits previous advice based on Bayesian networks. The authors reason whether an information provider is either truthful or untruthful by exploiting the correlation between current advice with previous advice. Although the proposed model, which is based on the theory of Bayesian network, appears to be an interesting approach that works very well for the ART testbed, its scalability is our main concern. In settings where millions of agents exist, expanding the network every time a new interaction between any of these agents occurs can result in a Bayesian network of a very large size. In contrast, Peng et al. [51] demonstrate that carefully defining the graph that depicts the interactions among autonomous parties can overcome significant complexity problems without severely compromising the quality of the information that is enclosed in the graph. In our approach we are interested in providing a model that exploits the consistency among the advice in such a way that provides strong incentives for the participant communities to be truthful while at the same time it can scale to the number of agents and to the number of the interactions between the communities. Our model encapsulates in a more sophisticated way characteristics similar to the one Hendrix et al. use (i.e., correlation among advice) and to the one that Peng et al. [51] propose (i.e., a more simplistic representation of the information that describes interactions that can allow us to reduce the computational cost without compromising the quality of the extracted information).

Last, we would like to note that as in many systems [22, 51, 76], which involves decisions based on information which relys on prior actions, design decisions have to be made in order to overcome the bootstrapping problem. For our selection mechanism, a simple way to solve this problem would be for a community to initially set the probability p to a high value and then slowly decrease p as the community gathers more information. Recall, the probability p represents the likelihood of collaborating with another community without using our selection mechanism.

Payment Distribution

Several payment or credit based approaches for motivating agents to behave well when they provide goods have been proposed. Li et al. [41] address the problem of message relaying in unstructured peer to peer networks. Each time a node relays a message and it proves to be in the path between the sender of the message and the recipient, it receives a payment (reward). Although the aim of the authors is to provide incentives for the peers to relay the messages, they do not tackle the problem of strategic relaying (i.e., dropping certain messages from certain sources). Consequently, the proposed approach is prone to a number of malicious behaviours that can range from isolating a node by selectively dropping the messages that it should either relay or answer to relay the message to "friendly nodes" instead to nodes that might create the shortest path to the final recipient.

Cheng and Vassileva [11] address the problem of overloading a peer-based article sharing system with low quality contributions and also consider a micropayment based approach. In their setting, each time a user uploads a link, it is awarded a number of c-points based on its reputability for giving high quality postings. The evaluation of the links is strongly related both to their quality and also to the number of users they have been seen by. In order to increase the visibility of a link, the users can use their c-points to sponsor it and thus to improve its ranking in the search engine. However, the authors do not address the problem of users strategically reporting their evaluations for other users' links in order to increase the ranking of their own results, and assume that all the users are truthful in providing their evaluations.

Golle et al. [20] propose an incentive-based model for peer to peer networks. The main idea is based on the principle to *charge the agents for every download and reward them for every upload*. An approach similar to this latter approach was proposed by Ma et al. [42, 43]. In this case their incentive based mechanism determines the proper transfer bandwidth allocation for a peer that requests a file, based on its connection type, its utility function and its contribution. Unlike the mechanism proposed by Golle et al. [20], where the utility is directly decreased each time the agent consumes the resources of the P2P network, in the mechanism proposed by Ma et al. [42, 43] the utility is only influenced indirectly by the decrement of the download bandwidth the next time the agent requests a file. Yu et al. [84] suggest a payment-based approach for addressing the problem of finding a node that can answer a query. In this approach each node a accumulates credits (payments) each time either answers a query or provides a referral to a node that will either answer the query or it will provide a referral. The latter credits are used when a node b is interested in querying other nodes.

One common characteristic of the latter three approaches is that they consider as input objectively countable data that occur from direct experience (e.g. Golle et al. [20] consider the size of the files downloaded, the disk space used etc, Ma et al. [42, 43] consider the amount of upload time, Yu et al. [84] consider the number of nodes that answered the query or provided a referral). As we discussed in Chapter 2, none of these approaches considers the quality of the exchanged data. Thus, their information comes from simply observing objectively countable information. In our setting the information that is being evaluated is subjective (since different communities might assign a different rating for an agent with a common behaviour based on the model they deploy), while it might also be the result of strategic reporting. The exchange of information takes place between self-interested communities that have strong incentives to lie, thus a more sophisticated payment approach is needed. For this reason, we proposed a payment function that considers the interpretation of a rating and distributes the payments based both on the accuracy of the rating and subjective differences between different evaluation models. We argue that our proposed approach also achieves a level of fairness since it distinguishes to a certain extent, potential efforts from malicious communities to manipulate the payment distribution.

Regan et al. [59] present a Bayesian-based approach to learn the evaluation function of an agent providing a rating, in order to make use of the information received. In our work we focused on determining the rewards that each community should receive with respect to the quality of the information it provided rather than learning the evaluation model each community deploys. A key contribution of our work, therefore, is that it promotes truthful reports, distributes the payments in a fair manner and provides incentive to communities to improve their evaluation model.

The payment function we proposed is motivated by the work on scoring rules [67]. *Scoring rules* are a framework for eliciting probabilistic information from agents. We differ from the formal definition of scoring rules in that we determine a value based on the community's declarations, whether the information was judged to be good or not (a binary event) and how it relates to the judging agent's valuation (a continuous variable). In other words, we consider both a continuous rating a community provided but also the binary interpretation of that rating inside the community. Scoring rules typically are either concerned with discrete events [92] or continuous settings [67], but not both. In our case, the information that is being scored represents an evaluation and not a probability of an event to happen. Probabilities convey a common knowledge information. For instance, a probability 0.5 of a binary event that a *red* or *blue* card will appear is commonly translated as: half the time a *blue* card will appear and half the time a *red* will appear. In contrast two communities that observed the same behaviour of an agent might assign two different ratings based on how strict or lenient the evaluation model they deploy is.

Strictly Properness is main property of scoring rules that we were interested in maintaining. The reason is that a mechanism which is based on a strictly proper scoring rule is incentive compatible. Our *i-Function* is strictly proper since it is uniquely maximized when a community reports a rating equal to the one the recipient community experiences. Thus, our payment mechanism is incentive compatible. Since the truthfulness of a rating depends both on the evaluation model a community follows and its strategy, we treat a deviation differently with respect to which decision it influences in the recipient community. For instance, consider:

- a community C_A that provided a rating $r_A = 0.7$ with a type equal to *Good*, and

- a community C_B that provided a rating equal to $r_B = 0.5$ with a type equal also to *Good.*

Assume that the recipient community C experiences a rating equal to 0.6 and the agent's type is *Good.* It is less likely that the community C_A intentionally deviated from the actual rating 0.6 than the community C_B. This is because a rating equal to 0.5 might be more likely to mislead the community C from accepting a good agent than a rating equal to 0.7. The community C_B might tried to misinform the community C because the agent might have limited resources, and thus if it joins the community C the communities C_B and C may have to share the resources of the agent.

Moreover, given that the rating the recipient community experiences (1^{st} dimension) and the rating an advisor community can provide (2^{nd} dimension) can take any value in the interval $[\alpha, \beta]$, and that the type of an agent can be either *Poor* or *Good* (3^{rd} dimension), it was crucial to clearly articulate the properties the *i-Function*, $I : \mathbb{R} \times \mathbb{R} \times \{\text{good}, \text{poor}\} \mapsto \mathbb{R}$, should follow in these three dimensions. Thus, in our work we carefully studied and suggested the properties the *i-Function* $I(x, \hat{r}, \theta)$ should follow in all of these three dimensions and we demonstrate that there is indeed a family of functions that satisfy these properties.

In the last few years scoring rules have started to become popular in the multiagent system literature. Zohar and Rosenschein [92] tackle the problem of information elicitation through the use of scoring rules. The first main difference of our approaches is that Zohar and Rosenschein propose a mechanism that relies on the assumption that the sellers do not intentionally try to sabotage the buyer. In our setting, this assumption does not hold since the selling agents (i.e., advisors) have strong incentives to lie. This is especially true for cases where the information that was requested refers to either a very reputable and trustworthy agent or to an agent with poor contributions. In the first case, as we have discussed earlier, communities can badmouth a good agent in order to prevent another competitor community from acquiring their good agent and in the latter case they can oversell an agent in order to get rid of it. Zohar and Rosenschein's proposed approach is also built on proper scoring rules. However, the authors do not aim to provide a new category of scoring rules but they rather prove that a robust incentive mechanism can be built by using scoring rules.

Furthermore, we differ from the above approach in the type of data that is being exchanged. Zohar and Rosenschein consider the exchange of probabilistic information. Thus, they assume a common interpretation of the exchanged information. As we discussed earlier, in our case we consider the exchange of information that represents the evaluation of an agent by a community. Given that each community might have a different evaluation model, communities may interpret the same rating differently. Consequently, the way a rating is interpreted by the provider community plays a crucial role in determining the payment the community should receive. Similar is the comparison with the work presented by Papakonstantinou et al. [50], since they also consider the exchange of probabilistic information whose interpretation appears to be common knowledge. Regarding the scoring rule part, Papakonstantinou et al. [50] focus on exploiting existing families of scoring rules rather than extending them. However, we do agree with the idea of a two phase incentive mechanism.

Jurca and Faltings [33] offer a side payment incentive compatible mechanism and prove that rational software agents under this mechanism will truthfully share their reputation information. As we discussed in the Chapter 2, Jurca and Faltings provide a payment mechanism that exploits an approach built on proper scoring rules. Their approach differs from ours since they consider that the probability distribution of the expected quality of the product $Pr(\theta)$ (where θ is the type that represents the quality of a product) is known while the real quality of the product a buyer experiences is unknown. Furthermore, they assume also that the probability distribution of the real quality given the answer of the buyer (i.e., $Pr(1|\theta)$ and $Pr(0|\theta)$ for all possible θ) is known. In our case, the expected quality of the agent is unknown while the real quality of the agent is considered known at the time the payment decision and distribution takes place. Recall, when a recipient community C^* experiences an agent and posts its rating about the agent on E's selling board, any community that buys this information will receive the latter information. If the community C^* tried to manipulate the payment distribution by reporting on E's selling board a rating different from the one it should, each time another community C' buys it, it will receive an inaccurate rating, and thus both the payment C' will receive and the probability that C' will ask it again in the future decrease. Furthermore, Jurca and Faltings consider the quality of a product but they do not consider towards which direction the latter quality influenced the buyer (i.e., to buy a product or not to buy). Moreover, in our approach we require both the type and the rating in order to cope with subjective differences which are the result of different evaluation models.

Miller et al. [46] address the problem of feedback elicitation and they also provide an incentive mechanism based on scoring rules. In their settings the decision of the score each peer will receive is defined based on the probability a rating reported is the actual rating other raters observed. In our case the rating, based on which the prediction is scored, is known.

Last, incentive compatibility for transactions that include purchase of items could be achieved through auction or auction-based mechanisms. However, the limitation that make auctions not applicable in our setting is the assumption that the auctioneer really wants to sell an item. In our case, if we consider a community that sells information about its agents as the auctioneer, this community might try to overprice its very good agents hoping that no other community will bid so high.

7.2 Trust and Reputation Modeling

Trust and reputation modeling along with the exploitation of trust and reputation information have been the subject of a wide area of applications in multiagent systems. Examples of these applications include:

- the evaluation of users in peer to peer networks [87],
- the evaluation of online feedback reporting [74],
- the monitoring of nodes in sensor networks [47] or in mobile ad hoc networks [8],
- task allocation [72],
- the decision of groups of agents that can collaborate to achieve common goals [66, 71],
- the evaluation of sellers and buyers in e-market places [77].

A limitation with reputation ratings is that they are highly related to the particular needs of a community. For example, a trustworthy agent might have low reputation (inside a community) because although it is willing to contribute there is no current interest in the services it offers. On the other hand a malicious agent can temporarily create a good reputation and once it has been accepted in another community become deceptive. A reliable reputation system should be able to ensure minimal impact when the second problem occurs. However, it is very difficult to cope with the case of a low reputation simply due to current lack of demand or due to incompatibility of services. For this reason, we argue that trust is a more comprehensive and objective metric for evaluating an agent, and consequently is a better evaluation metric for being exchanged between communities.

As Binmore and Dasgupta stated [6], we have significantly limited knowledge on how people acquire trust. Thus, we argue that trying to provide a general model of trust is very ambitious. For this reason, our trust model focuses on modeling trust in terms of promises. This follows the observation that in a number of cases people tend to reason about other people based on the extent they have honored a promise [13, 55]. In addition, we target communities where precise knowledge of the capabilities of an agent is crucial. Thus, we need to provide strong incentives to agents to truthfully report their promises and discourage them both from getting tempted to over-state or under-state them.

We are interested in designing a model that can be also applied in settings where the community might have to choose to follow a plan based on its current available resources, and thus is crucial both to know the exact services it has, and to provide strong incentives to agents to truthfully reveal the resources they can offer [4]. Thus, the trustworthiness of an agent should be only maximized when the agent delivers a contribution equal to its promise.

Furthermore, we argue that this is an important element since in any case where the available resources are used as an input in computationally intensive tasks, it is crucial to know a value as precisely as possible from the beginning. For instance, if the available resources are used in order for a community to decide to which task (or coalition) each agent will be better off assigned, the community has to know the exact amount of each of the services the agent can provide.[2] Since, task allocation [73] (or determining the structure of the coalitions among agents) is NP-Complete [65], re-computing the task allocation (or a coalition structure) each time an agent changes its behaviour might not be always feasible. Furthermore, another dimension is that frequent changes may lead to the creation of a feeling of instability among the rest of the users and thus discourage them from extending their participation.

In order to evaluate a promise we consider two factors. The first is how stable some agent is in keeping its promise (i.e., reliability) and the second is how good its contribution is (i.e., quality of delivered contribution). Our goal is to capture the behaviour of agents that can be either truthful or conservative or periodically overly optimistic or generally unstable. For example, consider a case where there are two agents a and b that are interested to purchase access to a webpage from a community. Assume that the community can only provide one access. The agent a promises to contribute \$5 dollars and agent b promises \$8. Assume now that:

- agent a has high reliability since it always delivers what it promises but in general its contributions are low, and
- agent b has low reliability, since it occasionally delivers less than it promised, however its contributions are in general high.

If the community grants access to the agent a, it will acquire \$5 but if it grants access to the agent b, and b does deliver its promise, then it will receive \$8. The community knows that a will honor its promise for sure but it also knows that the agent b might end up delivering even less than a. In order for the community to reason about which of these agents is a better choice, it has to consider a number of factors (e.g. if it is in an immediate need of the extra \$3 that a possible truthful report from b would result in). Thus, in some cases it might be a better choice to select agent a than to take the risk and select agent b and vice versa.

Summarizing, the key ideas around which we developed our model are:

[2] By the term “coalition” we refer to a group of agents that act in an coordinated manner in order to achieve a common goal.

1. measure the degree an agent honored its promises,
2. articulate the principles that a promise-based trust function should follow,
3. define the components a promised based trust model should include,
4. consider communities in which the exact knowledge of the resources their members contribute is crucial.

As we explained earlier in this chapter, we chose to develop a trust model that results in continuous instead of discete ratings, since we argue that this provides a more accurate and flexible tool to depict the trustworthiness of an agent. However, our model also considers continuous events as input (i.e., the agents have to declare their exact promise within a predefined range $[\alpha, \beta]$). This might raise the following question: "What if the agents are not requested to provide the exact promise, but an interval within the $[\alpha, \beta]$ range in which their promise will lie?" We argue that our proposed model can easily deal with this situation. This can be done by considering that an interaction is successful when an agent fullfills its promise (i.e., it delivers within the interval it promised). In this case, we can compute its score by using our trust function, by considering that the agent delivered the highest promise in the interval it declared.

Talwar et al. [74] also state the need for understanding a user's behaviour when evaluating the ratings or the information the user provides. However, they mainly focus on "exploring the factors that drive a user to submit a particular rating". Falcone and Castelfranchi [16] focus on understanding how trust functions for human users, as a motivation for designing a trust model. Our research differs, however, as our focus is in articulating the properties of contribution-based trust rather than the relationships between the parties that are trying to trust each other. Associating trust with reliability has been examined by other researchers [16, 44, 3], while considering the difference between what is expected (e.g. promised) and what is experienced (e.g. delivered) has been used in a number of other models [76, 77, 87]. Our model explicitly attempts to incorporate both the concepts of trust and of reliability.

Josang and Ismail [25] proposed the Beta Reputation System (also known as BRS) that exploits the beta function. The authors consider the number of cases an agent proved to be trustworthy and the number of cases it proved to be untrustworthy. Thus, they are focused on binary ratings. Similarly, Sen and Sajja [72] propose a trust model that evaluates the trustworthiness of agents that are requested to execute tasks on behalf of other agents by considering the number of times their performance was high and the number of times their performance was low. In contrast, we are interested in not only measuring whether an agent honored its promise but to what extent it did, and thus we consider continuous ratings. Teacy et al. [76] provide a model that essentially extends the Beta Reputation

System in coping with unfair ratings and thus it also considers binary ratings. Moreover, although the authors propose an approach that can protect their model from becoming unstable in terms of getting influenced by ratings that highly deviate by each other, it is still not clear to what extent it can prevent strategic behaviour.

Yu et al. [87] measured trust based on whether the reputation of an agent that encapsulates both direct experience and peer experience (when direct experience is not sufficient) exceeds a threshold. They consider continuous ratings but in their case they assume that each rating is assigned by the user. In contrast, in our work we provide a model that calculates this rating by comparing the promise an agent made with the action it delivered. Furthermore, Yu et al. provide a method for interpreting a set of trust ratings an agent was assigned to a general trust value while in our case we provide a method for determining the value of a trust rating. Regan et al. [58] focus on providing a model that evaluates the way advisors tend to provide advice. The authors are interested in developing a model that interprets the ratings different advisors provide with respect to the evaluation model they are using. In our case, we focus on direct experience in evaluating the trustworthiness of an agent rather than considering advice. Tran and Cohen [77] provide a reinforcement learning based technique for updating the reputation of a seller. Our first observation is that the authors essentially compute the trustworthiness a buyer assigns to a seller rather than its reputation since they only consider the direct interaction between the buyer and the seller. Furthermore, the proposed approach does not provide any incentives to prevent the sellers from following strategic behaviour and the principles of the proposed model are not stated. Finally, Kamvar et al. [34] suggested a solution for the problem of the pollution of a P2P network with inauthentic files, which results in reducing the performance and reliability of the network. However, the proposed approach does not take into consideration neither the information related to the size of the inauthentic files a peer downloads was not taken into consideration (e.g. downloading a big unauthentic file can be more annoying than downloading a small one, since in the first case the agent has to spend more resources) nor the case of an honest problem that resulted in providing an unauthentic file.

Clearly defining the principles that a trust model follows is crucial. This is because experimental evaluations are not always sufficient to clearly demonstrate the limitations of a trust model or the extent it can deal with strategic behaviour. Designing a generic trust model which is not prone to any strategic behaviour is challenging [38]. This is due to the fact that to this date, there is no model that can describe accurately how the humans reason about trustworthiness in every situation [52]. Thus, given that trust modeling in multiagent systems (where the agents represent the users) is essentially an effort to simulate humans' trust reasoning, is not clear how a general trust model for multiagent systems could be developed [16]. However, there are some patterns that can be identified in human trust decisions and that appear to be frequently in use. This has led to a perspective of trust that defines as trustworthiness the degree someone honors its promises or its commitments

[13]. However, when designing a promised-based trust model as well as any trust model for settings, where agents represent users, it is important to clearly articulate the principles the trust model follows since these principles provide the users with an important tool for interpreting the ratings they produce. For instance, knowing how a trust rating changes with respect to slight deviations of the behaviour an agent can provide useful information in interpreting a trust rating that was assigned to an agent.

Many trust models are based on probabilistic methods [22, 25, 31, 76, 87]. In our case, we agree with Schillo et al. [69] that *"trust is not an event in the sense of probability theory but rather a degree of how high some peer's honesty is estimated"*. Although probabilistic trust models can provide an important monitoring tool in applications where the results are exploited by experts we agree with Josang [30, 29] that *"a reputation system works when people can relate to it"* and we claim that this also holds for trust systems. Furthermore, Miller and Resnick [46] clearly state that a limitation of their proposed scoring model, which could be also used for reputation modeling, is *"Few raters will be willing or able to verify the mathematical properties of the scoring system proven in this paper, so it will be necessary to rely on outside attestations to ensure public confidence"*. Essentially they acknowledge the need for models that can be easily understood by the users. This is the main reason why e-market places are currently reluctant to implement complicated probabilistic models and simply report the number of positive and negative ratings a seller or a buyer receives. An average user lacks of deep knowledge in probability theory and thus, might not be able to understand a probabilistic model. Thus, an average user might not be able to relate to a probabilistic model since in order to relate to a model, he/she has first to understand it. On the other hand a more sophisticated model than simple summations (e.g. e-bay) is needed since these models are very susceptive to deception. For these reasons we argue that our non-probabilistic model, which is based on a set of principles that we believe can be easily explained to users who are represented by the agents, is valuable.

Chapter 8

Conclusions & Future Work

In this thesis we focused on peer-based communities. These are communities in which the services offered are provided by their participants. Our goals were:

1. to design a mechanism that can lead to the improvement of the services agents enjoy in these communities and,
2. to design a model to depict the trustworthiness of the agents within these communities.

In order to improve the services an agent enjoys in peer-based communities we need to improve the services other agents offer. Towards this goal we proposed a novel solution which allows communities to share the experience of their members with other communities. The experience of a community with an agent is captured in the evaluation rating of the agent within the community which can either represent the trustworthiness or the reputation of the agent. We argue that exchanging an agent's behaviour is the right way to improve the services the agent offers since it:

1. exploits the information that each community accumulates to allow other communities to decide whether to accept the agent while it also puts pressure on the agent to behave well since it is aware that any misbehaviour will be spread to the communities it might wish to join in the future.
2. can prevent the agent from overstretching itself among many communities, since this may lead the agent to provide very limited services to each of these communities due to its limited resources, and thus its trustworthiness and reputation might be compromised.

Allowing the communities to exchange information about the behaviour of their agents is a very challenging problem. The communities are self-interested and thus have strong incentives to play strategically. In order to prevent communities from manipulating the latter exchange, we considered a model in which each community requests information only from a subset of other communities and compensates them based on the quality of the information they provided. The compensation takes place through the use of a payment function which mainly considers the accuracy of the provided information but also treats inaccuracies differently when they appear to be the result of an inaccurate evaluation model and not the result of a strategic play.

The problems we addressed for facilitating the exchange of information between communities are:

1. *Advisor Selection Problem*: Select the set of advisor communities to consult.
2. *Payment Decision Problem*: Determine the payment a community should receive with respect to the quality of the information it provided.

The aim is to strengthen each community i) by encouraging the agents to be good contributors, since they can exploit good behaviour by carrying their reputation when joining new communities, and ii) by improving the quality of the information each community receives for deciding more carefully and accurately whether to accept an agent.

More specifically, our proposed solution for the *Advisor Selection Problem* allows the exchange of information between communities in such a way that the participant communities have strong incentives to provide truthful reports. In particular, we proposed a graph-based approach which focused on the exploitation of the consistency among the advice different advisors provide and in which honesty is a Bayes Nash Equilibruim. Although the idea of exploiting consistency appears to be an effective way to promote honesty, determining the maximum clique in a graph in addition to finding a subgraph of a graph G that is isomorphic to a graph H (known as the "subgraph isomorphism problem") are NP-Complete problems. At first glance this might appear to pose restrictions on the size of the graph that our approach can be applied. However, in a number of practical applications we expect the size of the consistency subgraph to be manageable either directly or through heuristics, since it is not unreasonable to assume an agent will be a member of only a few communities, thus limiting the size of the consistency subgraph. We would also like to note that there is work in the area of Constraint Satisfaction that can solve the subgraph isomorphism and maximal clique problem efficiently for satisfactory large order of graphs [60, 89].

Furthermore, for the *Payment Decision Problem* we proposed a novel scoring function which we used to determine the compensations the participant communities should receive with respect to both the accuracy and the interpretation of the information they provided. We presented the properties the scoring function should satisfy and then we provided examples of families of functions that could be used. Finally, we gave specific directions of possible criteria in selecting a specific instance of the latter families.

With respect to the *Trust Modeling Problem*, we agree that designing a generic trust model that can be applied to any environment is not feasible. Thus, we focus on providing a trust model and the principles it should follow for environments where communities request their agents to declare their anticipated contribution and for which the exact knowledge of the contribution of each agent is crucial. Although our model employs detailed mathematics, its concept can be intuitively explained to users (e.g. *QoC Property 1*: "The more you contribute, the more your contribution will be appreciated"). We feel this is valuable since, as we discussed, users of systems should be able to understand how their trust is modeled. In fact many companies are interested in having the users understand the limitations of the model in order to avoid the blame if the users get deceived [62] and is one reason, we believe, many companies only use very simple trust models like, for example, counting the number of positive and negative ratings. Moreover, in this work we provided a family of functions that satisfy these principles, we proved that our model is incentive compatible, and our empirical evaluation showed that our proposed model may provide an important improvement in the utility the agents enjoys with respect to other proposed models.

Concluding, the key contributions of this thesis are that we :

1. suggested addressing the problem of improving the services peer-based communities offer through the exchange of evaluation information of their agents,
2. suggested and proved that consistency exploitation can create strong incentives for promoting honesty in information exchange among self-interested participants,
3. provided an extension of the scoring rules to consider both a continuous rating and its binary interpretation,
4. suggested the principles a trust model that evaluates promises should follow and presented a mathematical model that satisfies them.

8.1 Future Work

In this section we provide some directions for future research with respect to the three problems we addressed.

Advisor Selection Problem

Although in a number of practical applications we expect the size of the consistency subgraph to be manageable either directly or through heuristics [89, 60], it would be still valuable to explore possible ways of replacing the cliques with other structures that will allow us to develop efficient algorithms for any order of graphs. Furthermore, it would be interesting to study to what extent our approach could address certain types of collusion since the colluding communities cannot have a clear view of the type of ties (i.e., edges in the *Consistency Graph*) the other members of the colluding group create. Collusion is a very challenging problem that emerges when different parties play strategically in collaborative or competitive environments. Our intuition is that under some circumstances, by using our proposed advisor selection mechanism, a strategic coordination of the colluding members is compromised even if at least one community has secretly built ties with communities outside this group.

Another interesting direction for future work is the area of *privacy preservation.* In our setting we have agents revealing private information about themselves to the new communities that they wish to join, and have communities sharing this information. We will need to decide what information should be kept public and what should be private and how to guard against inadvertently revealing information that should be kept private. In addition communities might be reluctant to reveal information about their agents. Furthermore, a major challenge in privacy preservation is to ensure that there is no unintentional leak of private information. For example, consider an agent that belongs to several communities and is going to reveal this information to a new community. Assume now that one of these communities is related to patients with chronic diseases, and the other is specialized in low sugar products. The knowledge of the participation of the agent in both of the communities can lead to the inference that the agent represents a diabetic user, information that the user might not wish to freely share with others.

An environment that is very similar to ours in regards to unintentionally revealing information is the ubiquitous computing environment. In a ubiquitous computing environment sensitive information can be revealed by considering the services a user has utilized. Hengartner and Steenkiste [23, 24] present a description of the main challenges in privacy preservation in ubiquitous computing environments. They identify the principles which should be taken into consideration when designing an access control mechanism for ubiquitous computing environments, and propose an architecture for allowing the flow of information among different nodes and entities that have access to information at different levels of granularity. Jiang and Landay [26] also discuss the different level of information granularity that different devices in a ubiquitous computing environment produce. In particular, they consider the problem of designing a privacy control system that can handle access to sensitive information which can be produced by a vast number of sensors in a ubiquitous computing environment.

The Payment Distribution Problem

Regarding the *Payment Distribution Problem*, we considered that the communities exchange the information they experience. It would be valuable to examine whether a community is better off reporting its real rating or a rating equal to the one other communities will experience. More specifically, our next step is to study such methods as those proposed by Regan et al. [59] as part of our effort to learn the way the selling communities model the ratings of their agents in order to both interpret the information that they provide but also to determine the ratings the buyer community should post on E's selling board. This is valuable since it might be more beneficial for a community to post on the trusted entity's E selling board a rating equal to the one that it believes other communities will experience rather than its real rating, given that the latter rating might provide more useful information to the recipient communities.

We believe that an important side effect of selling an estimate of the information that other communities will experience is that it will provide strong incentives towards the use of commonly acceptable evaluation models and thus might encourage a stronger collaboration in addressing the trust and reputation modeling problem. Furthermore, we would like to expand our proposed families of *i-Functions* in order to facilitate the selection of the most appropriate *i-Function* with respect to the particular requirements of the problem to which it is applied.

The Trust Modeling Problem

Regarding the *Trust Modeling Problem*, we are interested in extending the family of the QoC and RL functions in order to give communities the flexibility to find the optimal pair of functions along with the relation between them (i.e., multiplication, summations etc) that satisfies the properties of QoC, RL and Tr. In addition, we are interested in extending our empirical evaluation to more complex scenarios and study the performance of other instantiations of our proposed trust function.

Currently, we assumed that the significance of each service is fixed and predetermined by the community. In the future we plan to explore ways of using evaluation models in order to decide the significance of the combination of services an agent offers in conjunction with the amount of the contribution it is willing to provide for each service. For example an agent might be willing to provide 1 Gigabyte of disc space for other peer communities to store files and a download bandwidth of 512bkps, a second agent might provide 1 Gigabyte of disc space and 1gbps, while a third might be willing to provide 2 Gigabyte disc space and 2gbps of bandwidth but it can only be online 3 hours per day. Obviously, the significance of the bundle of the services each of the above three agents offers is different. Our vision is that a promising direction for determining the significance of the latter bundles is the area of combinatorial auctions.

Furthermore, it would be valuable to explore whether our trust function can be extended to environments in which the trustworthiness of an agent should be maximized in every case where the agent delivers at least what it promised. This can be done by slightly changing our trust function to get its maximum value if the delivered contribution is equal or more than the promised contribution. We can still retain the same QoC function to indicate the quality of the delivered contribution. Furthermore, the trust properties that refer to the interval $(\textit{promised contribution}, 1]$ can be ignored and we can simply treat the case of under-stating as equal to the case of truthfully reporting a promise. As we discussed in Chapter 6, a continuous rating can be easily converted to a discrete rating without loss of information by defining the intervals that correspond to each discrete value the rating can receive. An interesting direction for future research would be to design a model that defines the latter intervals.

Finally, an interesting dimension of trust is provided by Gambetta [19]. More specifically, Gambetta presents an analysis, from a sociological point of view, on the desirable amount of trust and he concludes that in some cases it is more desirable to "find the optimal mixture of cooperation and competition rather than decide at which extreme to converge". This observation might provide interesting insights on the set of parameters that should be taken into consideration for determining trustworthiness. For example, some interesting questions are:

- should the trustworthiness of an agent take into consideration the level of social interactions it has with other agents, and how?
- should the trustworthiness of an agent consider the level the agent contributes to or promotes the competitiveness of the environment?

Appendix A

APPENDICES

A.1 Diagrams for Chapter 6 Experiment 2

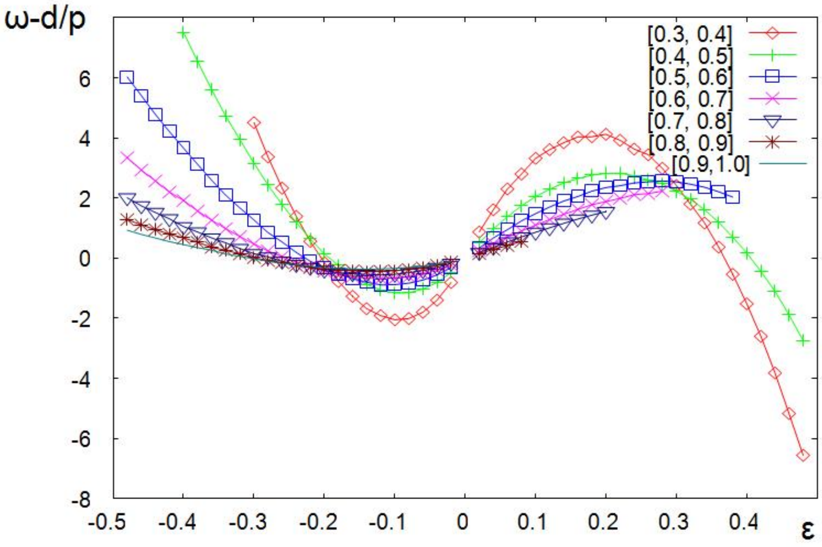


Figure A.1: Chapter 6-Experiment 2: The diagrams of $\omega - \frac{d}{p}$ for aggregation number equal to 100 for each interval in $\{[0.3, 0.4], ..., [0.9, 1]\}$

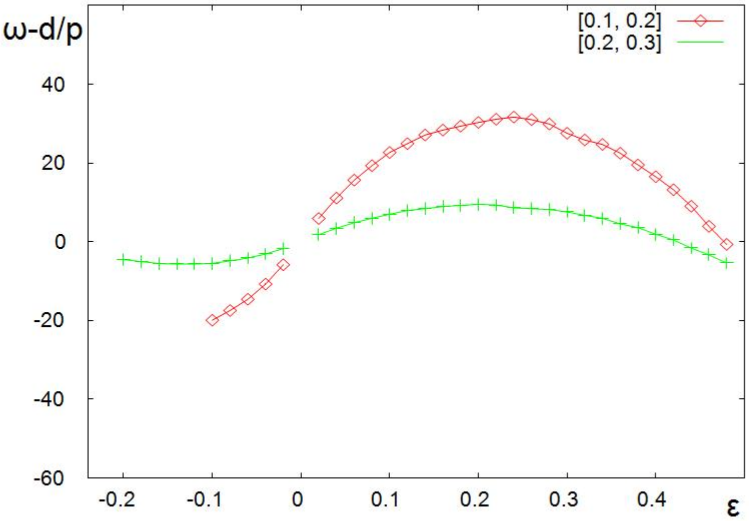


Figure A.2: Chapter 6-Experiment 2: The diagrams of $\omega - \frac{d}{p}$ for aggregation number equal to 100 for each interval in $\{[0.1, 0.2], [0.2, 0.3]\}$

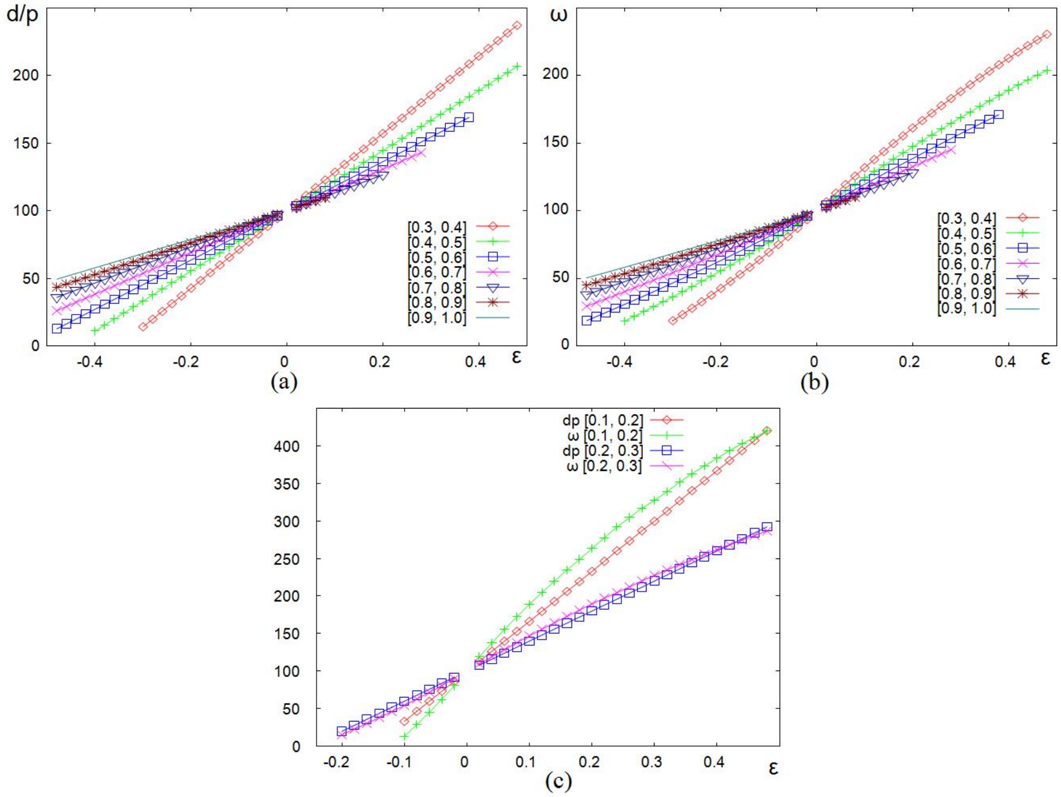
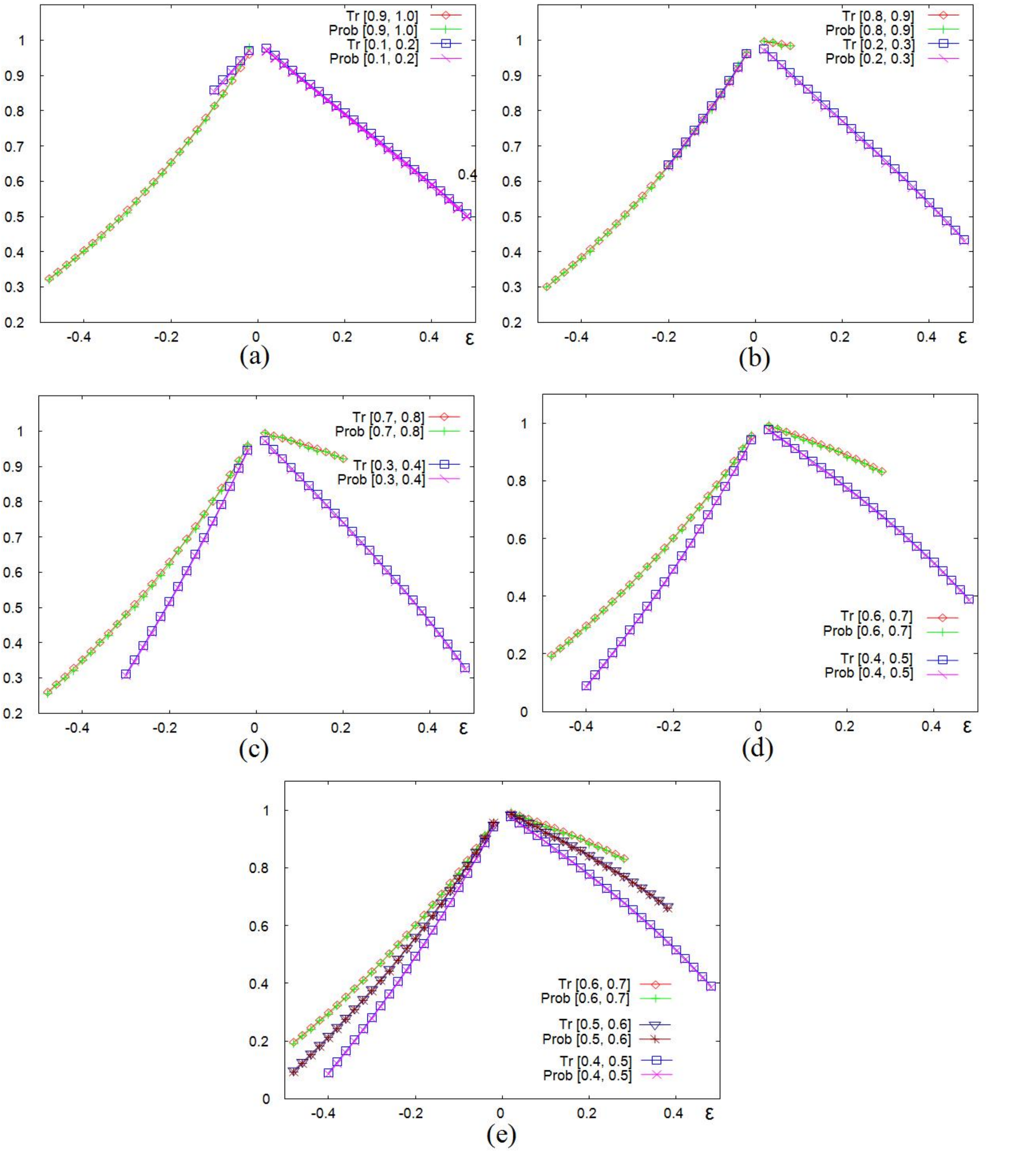


Figure A.3: Chapter 6-Experiment 2: The diagrams of ω and $\frac{d}{p}$ for aggregation number equal to 100 for each interval in $\{[0.1, 0.2], ..., [0.9, 1]\}$



(a) (b) (c) (d) (e)

Figure A.4: Chapter 6-Experiment 2: The average probability and average trust values per 100 ratings

Bibliography

[1] www.amazon.com.

[2] www.ebay.com.

[3] Aaditeshwar, Seth Jie Zhang, and Robin Cohen. A multi-disciplinary approach for recommending weblog messages. In *AAAI Workshop on Enhanced Messaging*, 2008.

[4] Mats Apelkrans and Anne Håkansson. Applying multi-agent system technique to production planning in order to automate decisions. In *KES-AMSTA '09: Proceedings of the Third KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 193–202, Berlin, Heidelberg, 2009. Springer-Verlag.

[5] Dipyaman Banerjee, Sabyasachi Saha, Sandip Sen, and Prithviraj Dasgupta. Reciprocal resource sharing in p2p environments. In *AAMAS '05: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 853–859, New York, NY, USA, 2005. ACM.

[6] Kenneth Binmore and Partha Dasgupta. Game theory: A survey. In *K. Binmore and P. Dasgupta (Eds) Economic Organizations as Games.* Basil Blackwell, 1986.

[7] Ronen I. Brafman, Carmel Domshlak, Yagil Engel, and Moshe Tennenholtz. Planning games. In *Proceedings of the Twenty First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 73–78, 2009.

[8] S. Buchegger and J. Y. Le Boudec. Self-policing mobile ad hoc networks by reputation systems. *Communications Magazine, IEEE*, 43(7):101–107, 2005.

[9] Cristiano Castelfranchi and Yao-Hua Tan. The role of trust and deception in virtual societies. In *HICSS '01: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 7*, page 7011, Washington, DC, USA, 2001. IEEE Computer Society.

[10] Alice Cheng and Eric Friedman. Sybilproof reputation mechanisms. In *P2PECON '05: Proceeding of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-peer Systems*, pages 128–132, New York, NY, USA, 2005. ACM Press.

[11] Ran Cheng and Julita Vassileva. Adaptive reward mechanism for sustainable online learning community. In *Proceeding of the 2005 Conference on Artificial Intelligence in Education*, pages 152–159, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press.

[12] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

[13] Partha Dasgupta. Trust as a commodity. In *Trust: Making and Breaking Cooperative Relations*, pages 49–72. Blackwell, 1998.

[14] Chrysanthos Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *EC '00: Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 150–157, New York, NY, USA, 2000. ACM.

[15] Chrysanthos Dellarocas. Analyzing the economic efficiency of ebay-like online reputation reporting mechanisms. In *EC '01: Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 171–179, New York, NY, USA, 2001. ACM Press.

[16] Rino Falcone and Cristiano Castelfranchi. The socio-cognitive dynamics of trust: Does trust create trust? In *Proceedings of the Workshop on Deception, Fraud, and Trust in Agent Societies held during the Autonomous Agents Conference*, pages 55–72, London, UK, 2001. Springer-Verlag.

[17] Michal Feldman, Kevin Lai, Ion Stoica, and John Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111, New York, NY, USA, 2004. ACM.

[18] Richard H. Franke and James D. Kaul. The hawthorne experiments: First statistical interpretation. *American Sociological Review*, 1978.

[19] Diego Gambetta. Can we trust trust? In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.

[20] Philippe Golle, Kevin Leyton-Brown, and Ilya Mironov. Incentives for sharing in peer-to-peer networksincentives for sharing in peer-to-peer networks. In *ACM Conference on Electronic Commerce*, pages 264–267, 2001.

[21] Garrett Hardin. The Tragedy of the Commons. *Science*, 162(3859):1243–1248, 1968.

[22] Philip Hendrix, Ya'akov Gal, and Avi Pfeffer. Learning whom to trust: Using graphical models for learning about information providers. In *AAMAS '09: Proceedings of the Eighth International Conference on Autonomous Agents and Multi-Agent Systems*, 2009.

[23] Urs Hengartner and Peter Steenkiste. Access control to information in pervasive computing environments. In *HOTOS'03: Proceedings of the 9th Conference on Hot Topics in Operating Systems*, pages 27–27, Berkeley, CA, USA, 2003. USENIX Association.

[24] Urs Hengartner and Ge Zhong. Distributed, uncertainty-aware access control for pervasive computing. In *Proc. of Fourth IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2007)*, pages 241–246. IEEE Computer Society, 2007.

[25] A. Josang R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Conference on Electronic Commerce*, 2002.

[26] Xiaodong Jiang and James A. Landay. Modeling privacy control in context-aware systems. *IEEE Pervasive Computing*, 1(3):59–63, 2002.

[27] Catholijn M. Jonker and Jan Treur. Formal analysis of models for the dynamics of trust based on experiences. In Francisco J. Garijo and Magnus Boman, editors, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99)*, volume 1647, pages 221–231, Berlin, 30– 2 1999. Springer-Verlag: Heidelberg, Germany.

[28] Audun Josang. Prospectives for online trust management, 2009. Working paper. Submitted to IEEE Transactions on Knowledge and Data Engineering.

[29] Audun Josang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.*, 43(2):618–644, 2007.

[30] Audun Jsang. Tutorial on trust and reputation systems. *IFIPTM2009*, 2009.

[31] Radu Jurca and Boi Faltings. An incentive compatible reputation mechanism. In *Proceedings of the IEEE Conference on E-Commerce*, pages 285–292, 2003.

[32] Radu Jurca and Boi Faltings. Enforcing Truthful Strategies in Incentive Compatible Reputation Mechanisms. In *Internet and Network Economics*, volume 3828, pages 268–277. Springer Verlag, 2005.

[33] Radu Jurca and Boi Faltings. Collusion-resistant, incentive-compatible feedback payments. In *EC '07: Proceedings of the 8th ACM Conference on Electronic commerce*, pages 200–209, New York, NY, USA, 2007. ACM.

[34] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM Press.

[35] Georgia Kastidou and Robin Cohen. Trust-oriented utility-based community structure in multi-agent systems. In J. Altmann D. Neumann, M. Baker and eds; Nielsen Books; O. Rama, editors, *Book Chapter in Economic Models and Algorithms for Distributed Systems*, pages 45–62. Spring Publisher, 2010.

[36] Georgia Kastidou, Robin Cohen, and Kate Larson. A graph-based approach for promoting honesty in community-based multiagent systems. In *the Eighth International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems (COIN'09), IJCAI'09*, 2009.

[37] Georgia Kastidou, Kate Larson, and Robin Cohen. Exchanging reputation information between communities: a payment-function approach. In *IJCAI'09: Proceedings of the 21st International Jont Conference on Artifical Intelligence*, pages 195–200, 2009.

[38] Reid Kerr and Robin Cohen. Smart cheaters do prosper: defeating trust and reputation systems. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 993–1000, Richland, SC, 2009.

[39] K. Lai, M. Feldman, Stoica, and J. Chuang. Incentives for cooperation in peer-to-peer networks. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[40] Kate Larson. *Fall 2008 Course Notes - CS886 Advanced Topics in Artificial Intelligence: Multiagent Systems.* Cheriton School of Computer Science, University of Waterloo, Canada.

[41] Cuihong Li, Bin Yu, and Katia Sycara. An incentive mechanism for message relaying in unstructured peer-to-peer systems. In *AAMAS '07: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1–8, New York, NY, USA, 2007. ACM.

[42] Richard T. B. Ma, C. M. Lee, John C. S. Lui, and David K. Y. Yau. Incentive p2p networks: a protocol to encourage information sharing and contribution. *SIGMETRICS Perform. Eval. Rev.*, 31(2):23–25, 2003.

[43] Richard T. B. Ma, Sam C. M. Lee, John C. S. Lui, and David K. Y. Yau. An incentive mechanism for p2p networks. In *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, pages 516–523, Washington, DC, USA, 2004. IEEE Computer Society.

[44] D. Harrison McKnight and Norman L. Chervany. The meanings of trust. Tech. Rep. MISRC Working Paper Series 96-04, 1996.

[45] Ruth M. Mickey, Olive Jean Dunn, and Virginia Clark. *Applied Statistics Analysis of Variance and Regression.* Wiley Series in Probability and Statistics, 2004.

[46] Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Manage. Sci.*, 51(9):1359–1373, 2005.

[47] Oly Mistry, Anil Gürsel, and Sandip Sen. Comparing trust mechanisms for monitoring aggregator nodes in sensor networks. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 985–992, 2009.

[48] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. Notions of reputation in multi-agent systems: A review. pages 280–287, Bologna, Italy, July 2002. First International Conference on Autonomous Agents and MAS, ACM.

[49] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1998.

[50] Athanasios Papakonstantinou, Alex Rogers, Enrico H. Gerding, and Nicholas R. Jennings. A truthful two-stage mechanism for eliciting probabilistic estimates with unknown costs. In *Proceeding of ECAI 2008*, pages 448–452. IOS Press, 2008.

[51] Wilbur Peng, William Krueger, Alexander Grushin, Patrick Carlos, Vikram Manikonda, and Michel Santos. Graph-based methods for the analysis of large-scale multiagent systems. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 545–552, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[52] Alireza Pourshahid and Thomas Tran. Modeling trust in e-commerce: an approach based on user requirements. In *ICEC '07: Proceedings of the Ninth International Conference on Electronic commerce*, pages 413–422, New York, NY, USA, 2007. ACM Press.

[53] Josep M. Pujol, Ramon Sangüesa, and Jordi Delgado. Extracting reputation in multi agent systems by means of social network topology. In *AAMAS '02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 467–474, 2002.

[54] Sarvapali D. Ramchurn, Dong Huynh, and Nicholas R. Jennings. Trust in multi-agent systems. *Knowl. Eng. Rev.*, 19(1):1–25, 2004.

[55] Sarvapali D. Ramchurn, Dong Huynh, and Nicholas R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19, 2004.

[56] Sarvapali D. Ramchurn, Claudio Mezzetti, Andrea Giovannucci, Juan A. Rodriguez-Aguilar, Rajdeep K. Dash, and Nicholas R. Jennings. Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty. *J. Artif. Int. Res.*, 35(1):119–159, 2009.

[57] Steve Reece, Alex Rogers, Stephen Roberts, and Nicholas R. Jennings. Rumors and reputation: Evaluating multi-dimensional trust within a decentralised reputation system. In *AAMAS '07: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1063–1070, New York, NY, USA, 2007. ACM.

[58] Kevin Regan, Robin Cohen, and Pascal Poupart. The advisor-pomdp: A principled approach to trust through reputation in electronic markets. In *Proceedings of Privacy, Security and Trust (PST05).*

[59] Kevin Regan, Pascal Poupart, and Robin Cohen. Bayesian reputation modeling in emarketplaces sensitive to subjectivity, deception and change. In *Proceedings of AAAI-06*, 2006.

[60] Jean Charles Regin. Solving the maximum clique problem with constraint satisfaction. In *5th international workshop on Integration of AI and OR Techniques in Constraing Programming for Combinatorial Optimization Problems*, 2003.

[61] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000.

[62] Paul Resnick, Richard Zeckhauser, John Swanson, and Kate Lockwood. The value of reputation on ebay: A controlled experiment. *Experimental Economics*, 9(2):79–101, June 2006.

[63] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, 2003.

[64] Jordi Sabater and Carles Sierra. Reputation and social network analysis in multi-agent systems. In *AAMAS '02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 475–482, New York, NY, USA, 2002. ACM.

[65] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohmé. Anytime coalition structure generation with worst case guarantees. In *AAAI*

'98/IAAI '98: Proceedings of the Fifteenth National/tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, pages 46–53, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[66] Davit Sarne and Sarit Kraus. The search for coalition formation in costly environments. *Lecture Notes in Computer Science*, (2782):117–136, 2003.

[67] Leonard Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.

[68] Paul Scerri, Yang Xu, Elizabeth Liao, Justin Lai, and Katia Sycara. Scaling teamwork to very large teams. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 888–895, Washington, DC, USA, 2004. IEEE Computer Society.

[69] Michael Schillo, Petra Funk, and Michael Rovatsos. Who can you trust: Dealing with deception. In *Proceedings of the Second Workshop on Deception, Fraud and Trust in Agent Societies, Seattle, USA*, pages 95–106, 1999.

[70] Sandip Sen and Dipyaman Banerjee. Monopolizing markets by exploiting trust. In *AAMAS '06: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1249–1256, New York, NY, USA, 2006. ACM.

[71] Sandip Sen, Indranil Goswami, and Stephane Airiau. Expertise and trustbased formation of effective coalitions: an evaluation of the art testbed. In *In Proceedings of the ALAMAS Workshop at the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 2006.

[72] Sandip Sen and Neelima Sajja. Robustness of reputation-based trust: boolean case. In *AAMAS '02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 288–293, New York, NY, USA, 2002. ACM.

[73] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, May 1998.

[74] Arjun Talwar, Radu Jurca, and Boi Faltings. Understanding user behavior in online feedback reporting. In *Proceedings of the ACM Conference on Electronic Commerce (EC'07)*, pages 134–142, 2007.

[75] Andrew S. Tanenbaum and Maarten van Steen. *Distributed Systems: Principles and Paradigms (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[76] W. T. Luke Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[77] Thomas Tran and Robin Cohen. Improving user satisfaction in agent-based electronic marketplaces by reputation modelling and adjustable product quality. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 828–835, Washington, DC, USA, 2004. IEEE Computer Society.

[78] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[79] Atul Singh Dan Wallach Tsuen-Wan Ngan, Animesh Nandi and Peter Druschel. On designing incentives-compatible peer-to-peer systems. In *In Proc. FuDiCo'04*, Bertinoro, Italy.

[80] Julita Vassileva, Silvia Breban, and Michael Horsch. Agent reasoning mechanism for long-term coalitions based on decision making and trust. *Computational Intelligence*, 18(4):583–595.

[81] Yao Wang and Julita Vassileva. Trust and reputation model in peer-to-peer networks. In *P2P '03: Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, page 150, Washington, DC, USA, 2003. IEEE Computer Society.

[82] Larry Wasserman. *All of Nonparametric Statistics (Springer Texts in Statistics)*. Springer, 2007.

[83] Bin Yu and Munindar P. Singh. A social mechanism of reputation management in electronic communities. In *CIA '00: Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace*, pages 154–165, London, UK, 2000. Springer-Verlag.

[84] Bin Yu and Munindar P. Singh. Detecting deception in reputation management. In *AAMAS '03: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 73–80, New York, NY, USA, 2003. ACM.

[85] Bin Yu and Munindar P. Singh. Incentive mechanisms for peer-to-peer systems. In *AP2PC*, pages 77–88, 2003.

[86] Bin Yu and Munindar P. Singh. Searching social networks. In *AAMAS '03: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 65–72, New York, NY, USA, 2003. ACM Press.

[87] Bin Yu, Munindar P. Singh, and Katia Sycara. Developing trust in large-scale peer-to-peer systems. In *Proceedings of First IEEE Symposium on Multi-Agent Security and Survivability*, pages 1–10, 2004.

[88] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic marketplaces. In *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, volume Track 8, 1999.

[89] Stephane Zampelli, Yves Deville, and Christine Solnon. *Solving Subgraph Isomorphism Problems with Constraint Programming.*

[90] Jie Zhang and Robin Cohen. Towards more effective e-marketplaces: A novel incentive mechanism. In *AAMAS Workshop on Trust in Agent Societies*, 2007.

[91] Jie Zhang, Robin Cohen, and Kate Larson. Theoretical validation and extended experimental support for a trust-based incentive mechanism for e-marketplaces. In *Eleventh International Workshop on Trust in Agent Societies*, 2008.

[92] Aviv Zohar and Jeffrey S. Rosenschein. Mechanisms for information elicitation. *Artif. Intell.*, 172(16-17):1917–1939, 2008.

[93] Aviv Zohar and Jeffrey S. Rosenschein. Adding incentives to file-sharing systems. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 859–866, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.