

The 19th European Conference on Artificial Intelligence

Proceedings

**Workshop on
Intelligent Agents and Technologies for e-Business
(IAT4EB)**

17 August 2010

Lisbon, Portugal

Ana Paula Rocha, LIACC, University of Porto, Portugal

Eugénio Oliveira, LIACC, University of Porto, Portugal

Virginia Dignum, Delft University of Technology, The Netherlands

Juan Antonio Rodriguez, IIIA-CSIC, Barcelona, Spain

Frank Dignum, Utrecht University, The Netherlands

Table of Contents

Multi-unit Auctions With Asymmetric Bidders.....	1
<i>Ioannis Vetsikas and Nicholas Jennings</i>	
Towards a Value-Sensitive System to support Agents in Norm Fulfillment and Enforcement.....	7
<i>Sietse Overbeek, Virginia Dignum and Yao-Hua Tan</i>	
An Agent Based Recommendation System for Tourism.....	13
<i>Joao Laranjeira, Joao Carneiro, Goreti Marreiros, Ricardo Santos and Carlos Ramos</i>	
Combining Loan Requests and Investment Offers in Peer-To-Peer Lending.....	19
<i>Luis Martinho and Luis Paulo Reis</i>	
In the Search of Better Deals Using Trust.....	25
<i>Maria Joana Urbano, Ana Paula Rocha and Eugénio Oliveira</i>	

Preface

The Workshop on Intelligent Agents and Technologies for e-Business (IAT4EB) provides a space for the discussion of the use of Artificial Intelligence and specifically Agent-based techniques dealing with interoperability issues in e-business environments. The main issues that the workshop focuses on are interaction, coordination, regulation and trust between agents that are part or represent different organizations.

A wide variety of electronic business scenarios and systems, as well as agent-based approaches to this subject, have been proposed in recent years. The primary goal of this workshop is to bring together the most recent and innovative research focusing on modelling, implementation, monitoring and evaluation of computational agents for e-business operations.

The topics covered by the accepted papers span well-established as well as emerging research areas, with a focus on Negotiation, Norm enforcement mechanisms, Recommendation systems and Trust measures.

In addition, the Workshop features an invited talk by Prof. Javier Vazquez-Salceda of the Universitat Politècnica de Catalunya on “Why shall we do this? Organizational awareness as an approach to create dynamic, flexible and context-aware eBusiness applications”.

We believe that IAT4EB 2010 is a good start attracting researchers working in the field of Distributed Systems that focus on modelling, implementation, monitoring and evaluation of computational agents for e-business operations. The first results have been encouraging. We hope for the continuation of this event in the future.

At this place we would like to thank the authors for their contributions and the reviewers for their excellent work of reviewing and selecting the best papers for presentation.

Ana Paula Rocha
Eugénio Oliveira
Virginia Dignum
Juan Antonio Rodriguez
Frank Dignum
(IAT4EB 2010 co-Chairs)

Multi-unit Auctions With Asymmetric Bidders

Ioannis A. Vetsikas and Nicholas R. Jennings¹

Abstract. In the existing work on auctions, a symmetric model of all bidders has been assumed: they all have the same risk attitude, possible valuations and spite (or competition). As this is not realistic in many real examples, we relax this assumption and extend the state of the art in multi-unit auctions to consider bidders of different types; each bidder type has a different risk attitude and distribution from which its valuation is drawn or different spite. We examine both the case when the participants' types are known and when they are not, and, for both cases, derive equilibrium bidding strategies in both m^{th} and $(m + 1)^{th}$ price sealed-bid auctions.

1 Introduction

Auctions have become commonplace and they are used to trade all kinds of commodity both between governments, companies, and private individuals. Game theory is widely used in such multi-agent scenarios, as a way to model and predict the behavior of bidders participating in these auctions. The scenarios normally analyzed in the auction literature (and other related disciplines) assume in almost all cases that the bidders participating in the auction are symmetric in the sense that they have their parameters (i.e. their valuations for the goods they bid to buy) drawn from the same prior distributions and have the same utility model. There is very little work that relaxes this assumption. More specifically, [6] and [4] compute equilibria for auctions with asymmetric bidders with different prior distributions from which their valuations are drawn, and an experimental evaluation is conducted in [2].

In this paper, we examine two important multi-unit auction scenarios that have been looked at in the auction literature: the case of bidders with any risk attitude, and of competitive bidders. In the existing work, a symmetric model of all bidders has been assumed; (i) they all have the same distribution from which their valuation is drawn and the same risk attitude thus using the same utility function, and (ii) they all have the same competitiveness respectively. For example, in [5] and [3], cases where agents are not risk neutral, but rather risk averse, are examined. In all instances, the agents are assumed to be risk averse in exactly the same way, and they all have the same utility function, which maps profit to utility in exactly the same way. In [1] and [9] a different kind of utility function is assumed; the bidders in this case not only wish to maximize their own profit, but they also wish to minimize the opponents' profit; these two goals are weighted by the agent's spite coefficient, which determines the relative importance assigned to these goals. In all this literature, the model of all agents is the same, in the sense that they all use the same valuation distribution function, the same utility function, and the same spite coefficient.

To extend these results, we introduce asymmetries in the bidders' models. However, unlike in [4], where the models of all bidders are

common knowledge, in this paper we examine not only this case, but also the one where we assume that each bidder only knows his own model: a bidder knows how competitive he is, but not how competitive the opponents are; he does know however that there is a certain chance associated with each opponent using a particular model (i.e. competition coefficient in this case).

This paper is organized as follows. In the next section, we formally present the model and the notation that will be used in this paper. Then, in section 3, we derive the systems of differential equations that characterize the Bayes-Nash equilibria that exist in the case of bidders with different risk attitudes and valuation distributions; first we analyze the case when the opponent models are not known and then the case when they are known to all participants. We also present a specific example to illustrate how to compute the equilibrium numerically; for the remaining cases we point the reader to the algorithm we presented in [10]. In section 4, we do the same for the case when the bidder spite (or competitiveness) is not the same for all bidders. Finally, we conclude.²

2 The Multi-Unit Auction Setting

In this section we formally describe the auction setting to be analyzed and define the objective function that the agents wish to maximize. We also give the notation that we use.

In particular, we will compute Bayes-Nash equilibria for sealed-bid auctions where $m \geq 1$ identical items are being sold; these equilibria will be defined by a set of strategies $g_{\alpha_i}(v)$, which map the agents' valuations v_i to bids b_i . These strategies are parameterized by a parameter α_i , which will indicate the model of agent i , i.e. his risk attitude and type of valuations or his spite. Thus we assume that two agents will use the same bidding strategy, if they have the same model (same parameter α_i). The final price is determined by the m^{th} price rule, according to which the top m bidders win one item each at a price equal to the m^{th} highest (last winning) bid respectively.

More specifically, we assume that N indistinguishable bidders (where $N \geq m$) participate in the auction and each has a private valuation (utility) v_i for acquiring any one³ of the traded items, which is known only to himself; these valuations are assumed to be independent drawn from a distribution with cumulative distribution function (cdf) $F_{\alpha_i}(v)$, which depends on the bidder's model α_i . Furthermore, we assume that $F_{\alpha_i}(v)$ has support in $[v_i^L, v_i^H]$, which means that

² We would like to point out that in the workshop paper [10] we presented some initial work analyzing the cases when the opponent models are not common knowledge. In this paper, we include some corrections and simplifications of that initial work, as well as the case when the models of all participants are known; furthermore, we extend the first case examined to include not only asymmetric risk attitudes, but also asymmetric valuations.

³ We make the assumption here that each bidder is interested in exactly one item; this is a usual assumption made in the analysis of multi-unit auctions, as the analysis even for self-interested risk-neutral bidders which are interested in purchasing multiple copies of the item is an open problem.

¹ University of Southampton, UK, email: {iv,nrj}@ecs.soton.ac.uk

$\forall v < v_i^L \vee v > v_i^H : F_{\alpha_i}'(v) = 0$.⁴ Let $\tilde{u}_i$ be the profit of agent i (i.e. $\tilde{u}_i = 0$, if it does not win an item, and $\tilde{u}_i = v_i - p_i$, if it does) and p_i is the total payment the agent must make to the auctioneer. The agents have varying risk attitudes. The possible risk attitudes belong to a family of utility functions $u_{\alpha_i}(\cdot)$, which are characterized by the type (model) α_i of each agent. Thus, we assume that the objective function (i.e. the total utility) that each agent tries to maximize depends only on his own gain $\tilde{u}_i$, and is equal to:

$$U_i = u_{\alpha_i}(\tilde{u}_i)$$

Some families of utility functions $u_{\alpha}(x)$ used widely in economics are: $u_{\alpha}(x) = x^{\alpha}$, $\alpha \in (0, 1)$ (CRRA), and $u_{\alpha}(x) = 1 - \exp(-\alpha x)$, $\alpha > 0$ (CARA); both characterize risk-averse bidders.

This model is the first scenario we analyze (in section 3), where we examine self-interested agents with different risk attitudes and valuation distributions. We examine two cases for this scenario. In the first, each agent has uncertainty not only for the opponents' valuations v_j , but also for their model (i.e. their parameters α_j ; they know the prior distribution $h(\alpha)$ from which each opponent parameter α is drawn, which is the probability that each participant is of a particular type α .⁵ So even though each agent knows only its own model, it can make a probabilistic inference on the possible opponent models. In the second case examined, the models of all participants are known.

In the second scenario, the agents are now risk-neutral and their valuations are drawn from the same distribution, but all agents are competitive, rather than self-interested. This means that they not only care about maximizing their profit, but also about minimizing the profit of the opponents. We define the objective function of each agent in the same way as in [1, 9]:

$$U_i = (1 - \alpha_i)\tilde{u}_i - \alpha_i \sum_{j \neq i} \tilde{u}_j$$

where $\alpha_i \in [0, 1]$ is a parameter called the *competition (or spite) coefficient*, which denotes the degree of competition of agent i ; the higher it is the more the agent cares about minimizing the opponent profit, rather than maximizing his own.

We also use the following additional notation in the proofs:

$$\Phi_k(x) = \sum_{i=0}^{k-1} C(N-1, i)x^{N-1-i}(1-x)^i \quad (1)$$

where the notation $C(n, k)$ is the total number of possible combinations of k items chosen from n . This formula is useful because, if $Z(x)$ is the probability distribution of any opponent's bid b_j , i.e. $Z(x) = \text{Prob}[b_j \leq x]$, and $B^{(k)}$ is the k^{th} order statistic of these bids of the opponents, then the distribution of $B^{(k)}$ is: [8]

$$\text{Prob}[B^{(k)} \leq x] = \Phi_k(Z(x)) \quad (2)$$

$\forall N, m$, such that $N \geq m$ the following equations hold: [9]

$$\Phi_m'(x) = (N-m)(\Phi_m(x) - \Phi_{m-1}(x))\frac{1}{x} \quad (3)$$

$$\Phi_m'(x) = m(\Phi_{m+1}(x) - \Phi_m(x))\frac{1}{1-x} \quad (4)$$

We will use equations 1 through 4 in the computation of the equilibria. To reduce the size of some equations in the proofs, let us also define:

$$\Delta\Phi_m(x) = \Phi_m(x) - \Phi_{m-1}(x) \quad (5)$$

We also need the following definition which will be used in the proofs of the equilibria when the models of all participants are common knowledge:

Definition 1 Given a set S (with all its elements being unique), let us define the k -subset $S^{(k)}$ of S to be the subset of its powerset 2^S whose elements have cardinality $k \leq |S|$. More formally:

$$S^{(k)} = \{s \in 2^S : |s| = k\}$$

⁴ For example, if $F_{\alpha_i}(v)$ is the uniform distribution $U[0, 1]$, then $v_i^L = 0$ and $v_i^H = 1$. If $F_{\alpha_i}(v)$ is such that v_i can take values in $[0, +\infty)$, then $v_i^L = 0$ and $v_i^H = +\infty$. We use these tight bounds to define the boundary conditions of some of the equilibria we will compute.

⁵ Thus $h(\alpha)$ is the probability that the opponent is of type α and this distribution is assumed unique, although the results can be extended to continuous distributions as well, as shown in [10].

For the specific case of the set $S = \{1, \dots, m\}$, let us define:

$$P_{k,m} = \{1, \dots, m\}^{(k)}$$

which is the set containing all the possible ways of selecting k different numbers out of the set of numbers 1 through m . Finally, we define the extensions of this definition:

$$P_{k,m}^{-\{i\}} = (\{1, \dots, m\} - \{i\})^{(k)}$$

$$P_{k,m}^{-\{i,j\}} = (\{1, \dots, m\} - \{i, j\})^{(k)}$$

which are the k -subset of all the numbers 1 through m without counting any subsets containing i (and i, j respectively).

3 Asymmetric Valuations and Risk Attitudes

In this section, we assume that agents have asymmetric valuations and risk attitudes. In section 3.1 we present the equations that characterize the equilibria when each bidder doesn't know the models of his opponents, while in section 3.2, we present the same analysis when the models of all participants is common knowledge. At the end of section 3.1, we give a specific example of how to derive the equilibrium strategies (for the case of theorem 1) numerically.

3.1 The Opponent Models Are Not Known

In this section, we assume that each agent has uncertainty not only for the opponents' valuations, but also for their models (i.e. risk attitudes and distributions of valuations). The possible risk attitudes and distributions of valuations belong to a family of functions, which are characterized by an one dimensional parameter α , which is drawn from a known probability distribution (h). We therefore assume that each agent i knows its own valuation v_i , risk attitude function $u_{\alpha_i}(\cdot)$ and the distributions $F_{\alpha_i}(v)$, as well as the distribution $h(\alpha)$ from which models of the opponents, meaning the risk attitude functions $u_{\alpha}(\cdot)$ and distributions $F_{\alpha}(v)$, are drawn. We assume that the number of possible models are λ , meaning that the possible models are characterized by $\alpha = \alpha_1, \dots, \alpha_{\lambda}$.

We initially present the system of equations that characterize the equilibrium and then show how to solve them.

Theorem 1 In the case of an m^{th} price sealed-bid auction with N participating bidders, in which each bidder i is interested in purchasing one unit of the good for sale with inherent utility (valuation) for that item equal to v_i , which is drawn from $F_{\alpha_i}(v)$, and has a risk attitude described by utility function $u_{\alpha_i}(\cdot)$, both of which describe his model α_i (where α_i are i.i.d. random variables drawn from distribution $h(\alpha)$), strategy $g_{\alpha_i}(v_i)$ constitutes a Bayes-Nash equilibrium, where $\zeta_{\alpha}(x) = g_{\alpha}^{-1}(x)$ is the solution of the system of differential equations:

$$\forall x, \alpha_i : (N-m) \sum_{\alpha=\alpha_1, \dots, \alpha_{\lambda}} F'(\zeta_{\alpha}(x)) \zeta_{\alpha}'(x) h(\alpha) = \frac{u_{\alpha_i}'(\zeta_{\alpha_i}(x) - x)}{u_{\alpha_i}(\zeta_{\alpha_i}(x) - x) - u_{\alpha_i}(0)} \sum_{\alpha=\alpha_1, \dots, \alpha_{\lambda}} F(\zeta_{\alpha}(x)) h(\alpha) \quad (6)$$

with boundary conditions: $g_{\alpha_i}(v_i^L) = v_i^L$ for all i such that $v_i^L = \min_j \{v_j^L\}$. There are λ possible bidder models characterized by parameter $\alpha = \alpha_1, \dots, \alpha_{\lambda}$.

Proof. We assume that the equilibrium strategy is described by functions $g_{\alpha}(v)$ which map the valuations v to bids for any of the possible risk attitude functions $u_{\alpha}(\cdot)$. We use this knowledge to determine the bids of the opponents and the expected profit that a bidder i gets from placing a bid equal to b_i . The distribution from which an opponent's bid b_j is drawn has cdf: $\text{Prob}[b_j \leq x | \alpha_j] = F_{\alpha_j}(g_{\alpha_j}^{-1}(x))$, when his risk attitude is described by function $u_{\alpha_j}(\cdot)$. Therefore, using Bayes' rule we compute this probability for any possible value of α_j :

$$\text{Prob}[b_j \leq x] = \sum_{\alpha=\alpha_1, \dots, \alpha_{\lambda}} F_{\alpha}(g_{\alpha}^{-1}(x)) h(\alpha) \quad (7)$$

The distribution of the k^{th} highest opponent bid $B^{(k)}$, as there are $(N - 1)$ opponents, is:

$$Prob[B^{(k)} \leq x] = \Phi_k \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F_\alpha(g_\alpha^{-1}(x))h(\alpha) \right) \quad (8)$$

where $\Phi_k(x)$ is given by equation 1.

We can now analyze the expected profit of bidder i . Let b_i be the bid that he places in the auction. We distinguish the following cases:

(i) If $b_i < B^{(m)}$, then bidder i is outbid and doesn't win any items, therefore his utility is $u_i = u_{\alpha_i}(0)$.

(ii) If $B^{(m)} \leq b_i \leq B^{(m-1)}$, then bidder i has placed the last winning bid. Thus the payment equals his bid and his utility is $u_i = u_{\alpha_i}(v_i - b_i)$. The probability of this case happening is: $Prob[B^{(m)} \leq b_i \leq B^{(m-1)}] = \Delta\Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F_\alpha(g_\alpha^{-1}(b_i))h(\alpha) \right)$.

(iii) If $B^{(m-1)} < b_i$, then bidder i is a winner, the payment is equal to bid $B^{(m-1)}$ and his utility is $u_i = u_{\alpha_i}(v_i - B^{(m-1)})$. Note that: $Prob[B^{(m-1)} \leq \omega] = \Phi_{m-1} \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F_\alpha(g_\alpha^{-1}(\omega))h(\alpha) \right)$.

The expected utility of bidder i , who places bid b_i , is:

$$\begin{aligned} EU_i(b_i) = & u_{\alpha_i}(0) \left(1 - \Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i))h(\alpha) \right) \right) \\ & + u_{\alpha_i}(v_i - b_i) \Delta\Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i))h(\alpha) \right) \\ & + \int_0^{b_i} u_{\alpha_i}(v_i - \omega) \frac{d}{d\omega} \left(\Phi_{m-1} \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(\omega))h(\alpha) \right) \right) d\omega \end{aligned} \quad (9)$$

The bid which maximizes this expected utility, is found by setting: $\frac{dEU_i}{db_i} = 0$. This becomes:

$$\begin{aligned} (u_{\alpha_i}(v_i - b_i) - u_{\alpha_i}(0)) \frac{d}{db_i} \Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i))h(\alpha) \right) \\ = u'_{\alpha_i}(v_i - b_i) \Delta\Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i))h(\alpha) \right) \end{aligned} \quad (10)$$

Thus using equation 3 to simplify equation 10, we derive:

$$(N-m) \frac{\frac{d}{db_i} \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i))h(\alpha) \right)}{\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i))h(\alpha)} = \frac{u'_{\alpha_i}(v_i - b_i)}{u_{\alpha_i}(v_i - b_i) - u_{\alpha_i}(0)}$$

This value b_i is equal to $b_i = g_{\alpha_i}(v_i)$, since it maximizes the expected utility $EU_i(b_i)$. Using this substitution, we derive the system of differential equations:

$$\forall v_i, \alpha_i : (N-m) \sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} \frac{F'_\alpha(g_\alpha^{-1}(g_{\alpha_i}(v_i)))}{g'_\alpha(g_\alpha^{-1}(g_{\alpha_i}(v_i)))} h(\alpha) = \quad (11)$$

$$\frac{u'_{\alpha_i}(v_i - g_{\alpha_i}(v_i))}{u_{\alpha_i}(v_i - g_{\alpha_i}(v_i)) - u_{\alpha_i}(0)} \sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F_\alpha(g_\alpha^{-1}(g_{\alpha_i}(v_i)))h(\alpha)$$

for all possible values of v_i, α_i . The boundary conditions come from the fact that a bidder with the lowest possible valuation that any bidder can have $v_i = v_i^L$ will always bid $b_i = v_i^L$.

Now, to simplify these equations we make the following substitutions:

(i) As the equations hold for all $\forall v_i, \alpha_i$, therefore, if we set a new variable $x = g_{\alpha_i}(v_i)$, which takes values in $x \in [g_{\alpha_i}(v_i^L), g_{\alpha_i}(v_i^H)]$, we transform the equations to the following:

$$\forall x, \alpha_i : (N-m) \sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} \frac{F'_\alpha(g_\alpha^{-1}(x))}{g'_\alpha(g_\alpha^{-1}(x))} h(\alpha) = \quad (12)$$

$$\frac{u'_{\alpha_i}(g_{\alpha_i}^{-1}(x) - x)}{u_{\alpha_i}(g_{\alpha_i}^{-1}(x) - x) - u_{\alpha_i}(0)} \sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(x))h(\alpha)$$

(ii) By setting $\zeta_{\alpha_i}()$ to be the inverse function of $g_{\alpha_i}()$, the equation becomes the system of equations 6. ■

Computing the Equilibrium Strategies The equations 6 seem quite complex. Thus, we show in this section how to solve them. We

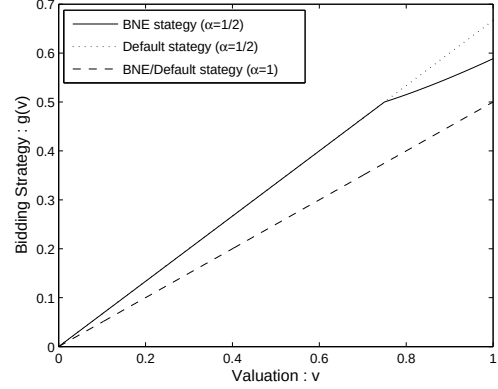


Figure 1. Equilibrium strategies $g(v)$ for an m^{th} price auction in which bidders use the CRRA utility function $u(x) = x^\alpha$ and there is a 50% chance that the opponents are risk neutral ($\alpha = 1$) and a 50% chance that they are risk averse with $\alpha = \frac{1}{2}$. Also included are the default equilibrium strategies (all the opponents have the same competition factor α as the bidder); in the case when $\alpha = 1$, the default strategy is the same of the BNE strategy, so it is not graphed separately. The valuations v are drawn from the uniform distribution $U[0, 1]$. The number of bidders is $N = 3$ and the number of items being sold is $m = 2$.

assume that α_i are ordered based on the value of v_i^L , meaning that we order them so that $v_1^L \leq \dots \leq v_\lambda^L$. This assumption is crucial for the following steps to work:

(i) In order for this system to have a solution it must be:

$$\frac{u'_{\alpha_1}(\zeta_{\alpha_1}(x) - x)}{u_{\alpha_1}(\zeta_{\alpha_1}(x) - x) - u_{\alpha_1}(0)} = \dots = \frac{u'_{\alpha_\lambda}(\zeta_{\alpha_\lambda}(x) - x)}{u_{\alpha_\lambda}(\zeta_{\alpha_\lambda}(x) - x) - u_{\alpha_\lambda}(0)} \quad (13)$$

This gives $(\lambda - 1)$ independent equations; differentiating each one of these gives us the following:

$$\zeta'_{\alpha_i}(x) = 1 + (\zeta'_{\alpha_1}(x) - 1) \quad (14)$$

$$\frac{u'_{\alpha_i}(\zeta_{\alpha_i}(x) - x)u'_{\alpha_1}(\zeta_{\alpha_1}(x) - x) - u''_{\alpha_1}(\zeta_{\alpha_1}(x) - x)(u_{\alpha_i}(\zeta_{\alpha_i}(x) - x) - u_{\alpha_i}(0))}{u'_{\alpha_1}(\zeta_{\alpha_1}(x) - x)u'_{\alpha_i}(\zeta_{\alpha_i}(x) - x) - u''_{\alpha_i}(\zeta_{\alpha_i}(x) - x)(u_{\alpha_1}(\zeta_{\alpha_1}(x) - x) - u_{\alpha_1}(0))}$$

which is used to substitute all $\zeta'_{\alpha_i}()$ with terms containing only $\zeta'_{\alpha_1}()$ in equation 13. Thus we derive a differential equation $\zeta'_{\alpha_1}()$ is equal to a function of $\zeta_{\alpha_i}()$, $\forall i$, where $\zeta_{\alpha_i}()$ can be computed from $\zeta_{\alpha_1}()$ using equation 13. This is solved by using the standard Runge-Kutta method, whose algorithm is presented in chapter 17 of [7], with one modification: the values of $\zeta_{\alpha_i}()$, $i = 2, \dots, \lambda$ are computed at each step from the values of $\zeta_{\alpha_1}()$ solving equation 13 using the Bisection Method; see chapter 9 of [7] for this algorithm.

(ii) Because in step 1, x is defined for $x \in [g_{\alpha_i}(v_i^L), g_{\alpha_i}(v_i^H)]$, we need be careful when $\zeta_{\alpha_i}(x) > v_i^H$ or $\zeta_{\alpha_i}(x) < v_i^L$ for any i . For such values, it is $F(\zeta_{\alpha_i}(x)) = 0$ and $F(\zeta_{\alpha_i}(x)) = 1$ respectively and also $F'(\zeta_{\alpha_i}(x)) = 0$. When performing the simplification of the previous step, we need to keep in mind this fact and that the equations 13 only hold for values of x such that $\zeta_{\alpha_i}(x) \in [v_i^L, v_i^H]$.

Example We give now a simple example of asymmetric risk attitudes. We examine an m^{th} price auction, with $N = 3$ bidders and $m = 2$ items for sale, where there are two possible models of bidders using the CRRA utility function $u_\alpha(x) = x^\alpha$, one where $\alpha = 1$ (risk neutral bidder) and another where $\alpha = 0.5$ (risk averse), both with probability 50%. In this example we have the following system of equations (obtained from equations 13 and 6 by setting $u_\alpha(x) = x^\alpha$ for $\alpha_i = 0.5, 1$ and probabilities $h(0.5) = h(1) = 0.5$):

$$\frac{1}{\zeta_1(x) - x} = \frac{0.5}{\zeta_{0.5}(x) - x} \quad (15)$$

$$\zeta'_1(x) + \zeta'_{0.5}(x) = \frac{0.5}{\zeta_{0.5}(x) - x} (\zeta_{0.5}(x) + \zeta_1(x)) \quad (16)$$

We present the equilibrium strategies in figure 1. It is interesting to note that the strategy for each asymmetric risk-averse bidder is, in this example, identical to the case when all his opponents are equally risk-averse (the symmetric bidder case). However, when the valuation is high enough that the risk-neutral opponents would never outbid the risk-averse bidders, the latter increase their bids at a much lower rate as the valuation increases. A similar effect is true for the risk-seeking bidders as well. In fact, we can prove this observation, for cases of bidders with identical valuation distribution functions.

For the case of an $(m+1)^{th}$ price auction, it is still a (weakly) dominant strategy to bid truthfully, i.e. use the same strategy as in the case when all the bidders use the same model:

Fact 1 *In the case of an $(m+1)^{th}$ price sealed-bid auction with N participating bidders, in which each bidder i is interested in purchasing one unit of the good for sale with inherent utility (valuation) for that item equal to v_i , and has a risk attitude described by utility function $u_{\alpha_i}(\cdot)$, it is a (weakly) dominant strategy to bid truthfully: $b_i = v_i$.*

This fact also holds for the case when the opponent models are common knowledge, which is examined in the next section.

3.2 Common Knowledge Of The Opponent Models

In this section we examine the same setting as in the previous one, with the difference that the models of all opponents are common knowledge to all participants.

Theorem 2 *Consider the same setting as that of theorem 1, with the difference that the models α_i of all bidders are now common knowledge. Then, strategy $g_{\alpha_i}(v_i)$ constitutes a Bayes-Nash equilibrium, where $\zeta_{\alpha}(x) = g_{\alpha}^{-1}(x)$ is the solution of the system of differential equations:*

$$\sum_{j=1}^N \frac{\zeta'_{\alpha_j}(x) F_{\alpha_j}(\zeta_{\alpha_j}(x))}{(u_{\alpha_i}(\zeta_{\alpha_i}(x) - x) - u_{\alpha_i}(0))} \sum_{s \in P_{m-1,N}^{-\{i,j\}}} \left(\prod_{\mu \notin s} F_{\alpha_{\mu}}(\zeta_{\alpha_{\mu}}(x)) \prod_{\mu \in s} (1 - F_{\alpha_{\mu}}(\zeta_{\alpha_{\mu}}(x))) \right) = \frac{u'_{\alpha_i}(\zeta_{\alpha_i}(x) - x)}{(u_{\alpha_i}(\zeta_{\alpha_i}(x) - x) - u_{\alpha_i}(0))} \sum_{s \in P_{m-1,N}^{-\{i\}}} \left(\prod_{j \notin s} F_{\alpha_j}(\zeta_{\alpha_j}(x)) \prod_{j \in s} (1 - F_{\alpha_j}(\zeta_{\alpha_j}(x))) \right) \quad (17)$$

with boundary conditions: $g_{\alpha_i}(v_i^L) = v_i^L$ for all i such that $v_i^L = \min_j \{v_j^L\}$.

Proof. Similar to the proof of theorem 1, we compute that the distribution from which the bid b_j of an opponent with model α_j is drawn has cdf: $Prob[b_j \leq x | \alpha_j] = F_{\alpha_j}(g_{\alpha_j}^{-1}(x))$. Now, bidder i faces $(N-1)$ opponents, which are the agents $\{1, \dots, N\} - \{i\}$. The distribution of the k^{th} highest bid $B_i^{(k)}$ among the bids of agent i 's opponents is $\Phi_k^i(x) = Prob[B_i^{(k)} \leq x]$. This is computed as:

$$\Phi_k^i(x) = \sum_{l=1}^k \sum_{s \in P_{l-1,N}^{-\{i\}}} \left(\prod_{j \notin s} F_{\alpha_j}(g_{\alpha_j}^{-1}(x)) \prod_{j \in s} (1 - F_{\alpha_j}(g_{\alpha_j}^{-1}(x))) \right) \quad (18)$$

The derivative of this equation is:

$$\frac{d\Phi_k^i(x)}{dx} = \sum_{j=1}^N \frac{d(F_{\alpha_j}(g_{\alpha_j}^{-1}(x)))}{dx} \sum_{s \in P_{k-1,N}^{-\{i,j\}}} \left(\prod_{\mu \notin s} F_{\alpha_{\mu}}(g_{\alpha_{\mu}}^{-1}(x)) \prod_{\mu \in s} (1 - F_{\alpha_{\mu}}(g_{\alpha_{\mu}}^{-1}(x))) \right) \quad (19)$$

Using the same reasoning as in theorem 1, we compute the expected utility of bidder i , who places bid b_i , as:

$$EU_i(b_i) = u_{\alpha_i}(0)(1 - \Phi_m^i(b_i)) + u_{\alpha_i}(v_i - b_i)(\Phi_m^i(b_i) - \Phi_{m-1}^i(b_i)) + \int_0^{b_i} u_{\alpha_i}(v_i - \omega) \frac{d}{d\omega} (\Phi_{m-1}^i(b_i)) d\omega \quad (20)$$

The bid which maximizes this expected utility, is found by setting: $\frac{dEU_i}{db_i} = 0$. This becomes:

$$(u_{\alpha_i}(v_i - b_i) - u_{\alpha_i}(0)) \frac{d}{db_i} \Phi_m^i(b_i) = u'_{\alpha_i}(v_i - b_i) (\Phi_m^i(b_i) - \Phi_{m-1}^i(b_i)) \quad (21)$$

This value b_i is equal to $b_i = g_{\alpha_i}(v_i) \Leftrightarrow v_i = g_{\alpha_i}^{-1}(b_i)$, since it maximizes the expected utility $EU_i(b_i)$. Using this substitution, we derive:

$$\sum_{j=1}^N \frac{d(F_{\alpha_j}(g_{\alpha_j}^{-1}(b_i)))}{db_i} \sum_{s \in P_{m-1,N}^{-\{i,j\}}} \left(\prod_{\mu \notin s} F_{\alpha_{\mu}}(g_{\alpha_{\mu}}^{-1}(b_i)) \prod_{\mu \in s} (1 - F_{\alpha_{\mu}}(g_{\alpha_{\mu}}^{-1}(b_i))) \right) = \frac{u'_{\alpha_i}(g_{\alpha_i}^{-1}(b_i) - b_i)}{(u_{\alpha_i}(g_{\alpha_i}^{-1}(b_i) - b_i) - u_{\alpha_i}(0))} \sum_{s \in P_{m-1,N}^{-\{i\}}} \left(\prod_{j \notin s} F_{\alpha_j}(g_{\alpha_j}^{-1}(b_i)) \prod_{j \in s} (1 - F_{\alpha_j}(g_{\alpha_j}^{-1}(b_i))) \right) \quad (22)$$

Defining $\zeta_{\alpha_i}(x) = g_{\alpha_i}^{-1}(x)$, which means that $\zeta(\cdot)$ are the inverse of $g(\cdot)$, we derive the system of differential equations 17 for all possible values of $x = b_i$ and for every agent i with model α_i . The boundary conditions come from the fact that a bidder with the lowest possible valuation that any bidder can have $v_i = v_i^L$ will always bid $b_i = v_i^L$. ■

Computing the solution of this system of differential equations as well as the systems characterizing the equilibria of the next section are done in the manner we described in [10].

4 Asymmetric Competitiveness

In this section, we assume that agents have asymmetric competitiveness (spite). In section 4.1 we present the equations that characterize the equilibria when each bidder doesn't know the models of his opponents, while in section 4.2, we present the same analysis when the models of all participants are common knowledge to all participants.

4.1 The Opponent Models Are Not Known

In this section, we assume that each agent has uncertainty not only for the opponents' valuations, but also for how competitive they are. The competitiveness of an agent is characterized by his competition coefficient α_i , which takes values in $[0, 1]$, which is drawn from a known probability distribution $h(\alpha)$. We therefore assume that each agent i knows its own valuation v_i and competition coefficient α_i , and also the distributions F and h from which the valuations v_i and competition coefficients α_i of the other agents are drawn. We assume that the number of possible models are λ , meaning that the possible bidder types have competitiveness $\alpha = \alpha_1, \dots, \alpha_{\lambda}$.

Theorem 3 *In the case of an m^{th} price sealed-bid auction with N participating risk-neutral bidders, in which each bidder i is interested in purchasing one unit of the good for sale with inherent utility (valuation) for that item equal to v_i , and has a competition coefficient α_i , where v_i and α_i are i.i.d. random variables drawn from distributions $F(v)$ and $h(\alpha)$ respectively, strategy $g_{\alpha_i}(v_i)$ constitutes a Bayes-Nash equilibrium, where $\zeta_{\alpha_i}(x) = g_{\alpha_i}^{-1}(x)$ is the solution of the system of differential equations:*

$$\frac{1 - \alpha_i m}{N - m} \sum_{\alpha = \alpha_1, \dots, \alpha_{\lambda}} F(\zeta_{\alpha}(x)) h(\alpha) = \quad (23)$$

$$\sum_{\alpha = \alpha_1, \dots, \alpha_{\lambda}} \zeta'_{\alpha}(x) F'(\zeta_{\alpha}(x)) (\zeta_{\alpha}(x) - x - \alpha_i \zeta_{\alpha}(x) + \alpha_i \zeta_{\alpha}(x)) h(\alpha)$$

with boundary conditions: $g_{\alpha_i}(v_i^L) = v_i^L, \forall \alpha_i$.

Proof. We assume that the equilibrium strategy is described by functions $g_{\alpha}(v)$ which map the valuations v to bids for any of the competition factors α . We will use this knowledge to determine the bids of

the opponents and the expected profit that a bidder i gets from placing a bid equal to b_i . The distributions of any one opponent bid b_j and of the k^{th} highest opponent bid $B^{(k)}$ are given from equations 7 and 8 (use the same reasoning as in theorem 1). Now, bidder i bids b_i , the bid that maximizes his objective function on expectation.

Let C be the sum (on expectation) of the $(m-1)$ opponent valuations that produced the top $m-1$ winning bids. Since in all cases that we will examine, whether bidder i wins or not, we know that the opponents with the top $(m-1)$ bids will each win an item, we know that they will gain this amount C from doing so. This value is a constant and does not depend on the bid b_i . We will mostly ignore this term in the rest of the computations.

Depending on the bid b_i , we need to consider the following cases: (i) When $B^{(m)} > b_i$, bidder i does not win any item and the closing price is $B^{(m)}$. Therefore bidder i 's gain is 0 and the opponents make a gain from gaining an extra item (the m^{th}), in addition to the $(m-1)$ items that they always win (this was counted in the constant value C). We must compute the expected gain obtained by getting this extra item. Let us assume that the actual value of $B^{(m)} = x$. This is equal to a bid submitted by an agent (w.l.o.g. assume this is agent j). Then $b_j = x$ and we want to find out the expectation of the value of the valuation v_j that generated this bid for all possible values of α_j . Let us denote this by $EV(x) = E(v_j | b_j = x)$. For a particular value of α_j , i.e. when $\alpha_j = \alpha$, we know that $v_j = g_\alpha^{-1}(x)$ and this happens with probability $Prob[b_j = x | \alpha_j = \alpha] = \frac{d}{dx} F(g_\alpha^{-1}(x))$. Using Bayes' rule we can compute the value of $E(v_j | b_j = x)$ being equal to:

$$EV(x) = \frac{\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} g_\alpha^{-1}(x) \frac{d}{dx} F(g_\alpha^{-1}(x)) h(\alpha)}{\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} \frac{d}{dx} F(g_\alpha^{-1}(x)) h(\alpha)} \quad (24)$$

They also must make total payments of $mB^{(m)}$. The total additional⁶ expected utility for bidder i in this case is hence:

$$\Delta U_1 = \alpha_i \int_{b_i}^{\infty} (m\omega - EV(\omega)) \frac{d}{d\omega} \left(\Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(\omega)) h(\alpha) \right) \right) d\omega \quad (25)$$

(ii) When $B^{(m-1)} > b_i \geq B^{(m)}$, bidder i wins an item and the closing price is b_i . Therefore bidder i 's gain is $v_i - b_i$ and the opponents pay $(m-1)b_i$ for the items that they win. The total additional expected utility for bidder i is:

$$\Delta U_2 = ((1-\alpha_i)(v_i - b_i) + \alpha_i(m-1)b_i) \Delta \Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i)) h(\alpha) \right) \quad (26)$$

(iii) When $b_i \geq B^{(m-1)}$, bidder i wins an item and the closing price is $B^{(m-1)}$. Therefore bidder i 's gain is $v_i - B^{(m-1)}$ and the opponents must pay $(m-1)B^{(m-1)}$ for the items that they purchase. The total additional expected utility for bidder i in this case is:

$$\Delta U_3 = \int_0^{b_i} ((1-\alpha_i)(v_i - \omega) + \alpha_i(m-1)\omega) \frac{d}{d\omega} \left(\Phi_{m-1} \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(\omega)) h(\alpha) \right) \right) d\omega \quad (27)$$

The total expected utility for bidder i when considering all possibilities is therefore: $EU_i(b_i) = -\alpha_i C + \Delta U_1 + \Delta U_2 + \Delta U_3$. To find the value of v_i that maximizes the expected utility $EU_i(b_i)$, we set $\frac{dEU_i(b_i)}{db_i} = 0$. We then get:

$$(1 - \alpha_i m) \Delta \Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i)) h(\alpha) \right) = \quad (28)$$

$$(v_i - b_i - \alpha_i v_i + \alpha_i EV(b_i)) \frac{d}{db_i} \left(\Phi_m \left(\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i)) h(\alpha) \right) \right)$$

By using equation 3 to simplify equation 28, we get:

$$\frac{1 - \alpha_i m}{N - m} \sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i)) h(\alpha) = \quad (29)$$

⁶ We mean additional to the fact that the agent always loses utility αC , since its opponents always gain a value C from the top $(m-1)$ items.

$$(v_i - b_i - \alpha_i v_i + \alpha_i EV(b_i)) \sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} \frac{d}{db_i} F(g_\alpha^{-1}(b_i)) h(\alpha)$$

Since strategy $g_\alpha(v)$ gives the equilibrium strategy, then it must be the case that the value of b_i that maximizes the total utility is given by $g_\alpha(v)$, i.e. that $b_i = g_\alpha(v_i) \Leftrightarrow v_i = g_\alpha^{-1}(b_i)$. Using this fact and equation 24 to substitute in equation 29, we get:

$$\frac{1 - \alpha_i m}{N - m} \sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(g_\alpha^{-1}(b_i)) h(\alpha) = \quad (30)$$

$$\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} \frac{F'(g_\alpha^{-1}(b_i))}{g_\alpha'(g_\alpha^{-1}(b_i))} (g_\alpha^{-1}(b_i) - b_i - \alpha_i g_\alpha^{-1}(b_i) + \alpha_i g_\alpha^{-1}(b_i)) h(\alpha)$$

Setting $\zeta_{\alpha_i}(x) = g_{\alpha_i}^{-1}(x)$, we derive the system of differential equations 23 for all possible values of $x = b_i$ and for every agent i with model α_i . We select the boundary condition $g_\alpha(v_i) = v_i$, based on the fact that this boundary condition holds when all the agents have the same competition factor α , $\forall \alpha$. (see [9]) ■

Theorem 4 In the case of an $(m+1)^{th}$ price sealed-bid auction with N participating risk-neutral bidders, in which each bidder i is interested in purchasing one unit of the good for sale with inherent utility (valuation) for that item equal to v_i , and has a competition coefficient α_i , where v_i and α_i are i.i.d. random variables drawn from distributions $F(v)$ and $h(\alpha)$ respectively, strategy $g_{\alpha_i}(v_i)$ constitutes a Bayes-Nash equilibrium, where $\zeta_{\alpha_i}(x) = g_{\alpha_i}^{-1}(x)$ is the solution of the system of differential equations:

$$\forall v_i, \alpha_i : -\alpha_i \left(1 - \sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} F(\zeta_\alpha(x)) h(\alpha) \right) = \quad (31)$$

$$\sum_{\alpha=\alpha_1, \dots, \alpha_\lambda} \zeta'_\alpha(x) F'(\zeta_\alpha(x)) (g_\alpha^{-1}(x) - x - \alpha_i g_\alpha^{-1}(x) + \alpha_i \zeta_\alpha(x)) h(\alpha)$$

with boundary conditions: $g_{\alpha_i}(v_i^H) = v_i^H, \forall \alpha_i$.

Proof. This proof as well as the proof of theorem 6 are omitted due to space. ■

4.2 Common Knowledge Of The Opponent Models

In this section we examine the same setting as in the previous one, with the difference that the models of all opponents are common knowledge. We initially examine the setting where an m^{th} price auction is used:

Theorem 5 Consider the same setting as that of theorem 3, with the difference that the models α_i of all bidders are now common knowledge. Then, strategy $g_{\alpha_i}(v_i)$ constitutes a Bayes-Nash equilibrium, where $\zeta_\alpha(x) = g_\alpha^{-1}(x)$ is the solution of the system of differential equations:

$$\sum_{j=1, j \neq i}^N \zeta'_{\alpha_j}(x) F'(\zeta_{\alpha_j}(x)) (\zeta_{\alpha_i}(x) - x - \alpha_i \zeta_{\alpha_i}(x) + \alpha_i \zeta_{\alpha_j}(x)) \cdot$$

$$\sum_{s \in P_{m-1, N}^{-\{i, j\}}} \left(\prod_{\substack{\mu \notin s \\ \mu \neq i, j}} F(\zeta_{\alpha_\mu}(x)) \prod_{\mu \in s} (1 - F(\zeta_{\alpha_\mu}(x))) \right) = \\ (1 - \alpha_i m) \sum_{s \in P_{m-1, N}^{-\{i\}}} \left(\prod_{\substack{j \notin s \\ j \neq i}} F(\zeta_{\alpha_j}(x)) \prod_{j \in s} (1 - F(\zeta_{\alpha_j}(x))) \right) \quad (32)$$

with boundary conditions: $g_{\alpha_i}(v_i^L) = v_i^L, \forall \alpha_i$.

Proof. Similar to the proof of theorem 3, we compute that the distribution from which the bid b_j of an opponent with model α_j is drawn has cdf: $Prob[b_j \leq x | \alpha_j] = F(g_{\alpha_j}^{-1}(x))$. Now, bidder i faces $(N-1)$ opponents, which are the agents $\{1, \dots, N\} - \{i\}$. The distribution of the k^{th} highest bid $B_i^{(k)}$ among the bids of agent i 's

opponents is $\Phi_k^i(x) = \text{Prob}[B_k^{(k)} \leq x]$, which is given by equation 18, and the pdf of this distribution by equation 19. We compute the expected utility of bidder i , who places bid b_i , as:
 $EU_i(b_i) = -\alpha_i C + \Delta U_1 + \Delta U_2 + \Delta U_3$, where:

$$\Delta U_1 = \alpha_i \int_{b_i}^{\infty} \left(m\omega \frac{d}{d\omega} \Phi_m^i(\omega) - \right. \quad (33)$$

$$\left. \sum_{j=1, j \neq i}^N g_{\alpha_j}^{-1}(\omega) \frac{d(F(g_{\alpha_j}^{-1}(\omega)))}{d\omega} \sum_{s \in P_{m-1, N}^{-\{i, j\}}} \left(\prod_{\mu \notin s} F(g_{\alpha_\mu}^{-1}(\omega)) \prod_{\mu \in s} (1 - F(g_{\alpha_\mu}^{-1}(\omega))) \right) \right) d\omega$$

$$\Delta U_2 = ((1 - \alpha_i)(v_i - b_i) + \alpha_i(m - 1)b_i) (\Phi_m^i(b_i) - \Phi_{m-1}^i(b_i)) \quad (34)$$

$$\Delta U_3 = \int_0^{b_i} ((1 - \alpha_i)(v_i - \omega) + \alpha_i(m - 1)\omega) \frac{d}{d\omega} \Phi_{m-1}^i(\omega) d\omega \quad (35)$$

In order to compute equations 34 and 35, we used the same reasoning as in theorem 3; these cover that cases that the bid b_i of bidder i is respectively the m^{th} highest bid and higher than the m^{th} bid, and therefore, in both cases, the bidder wins an item. Equation 33 represents the case that b_i is less than the m^{th} highest opponent bid in which case ΔU_1 is α_i times the difference of the expected payments made by the opponents minus the expected gain of the opponent who placed the m^{th} highest bid from getting the item; the first term is computed as in theorem 3. The second term is computed by not only examining the probability of the m^{th} highest opponent bid being equal to $\omega < b_i$, but also conditional on the fact that this bid was placed by a particular opponent j ; we account for all possible opponents j and thus derive the second term of equation 33.

To find the value of v_i that maximizes the expected utility $EU_i(b_i)$, we set $\frac{dEU_i(b_i)}{db_i} = 0$. We then get:

$$(v_i - b_i - \alpha_i v_i) \frac{d}{db_i} \Phi_m^i(b_i) + \alpha_i \cdot$$

$$\sum_{j=1, j \neq i}^N g_{\alpha_j}^{-1}(b_i) \frac{d(F(g_{\alpha_j}^{-1}(b_i)))}{db_i} \sum_{s \in P_{m-1, N}^{-\{i, j\}}} \left(\prod_{\mu \notin s} F(g_{\alpha_\mu}^{-1}(b_i)) \prod_{\mu \in s} (1 - F(g_{\alpha_\mu}^{-1}(b_i))) \right) db_i$$

$$= (1 - \alpha_i m) (\Phi_m^i(b_i) - \Phi_{m-1}^i(b_i))$$

This value b_i is equal to $b_i = g_{\alpha_i}(v_i) \Leftrightarrow v_i = g_{\alpha_i}^{-1}(b_i)$, since it maximizes the expected utility $EU_i(b_i)$. Using this substitution and using equations 18 and 19, we derive:

$$\sum_{j=1, j \neq i}^N (g_{\alpha_i}^{-1}(b_i) - b_i - \alpha_i g_{\alpha_i}^{-1}(b_i) + \alpha_i g_{\alpha_j}^{-1}(b_i)) \frac{d(F(g_{\alpha_j}^{-1}(b_i)))}{db_i} \cdot$$

$$\sum_{s \in P_{m-1, N}^{-\{i, j\}}} \left(\prod_{\mu \notin s} F(g_{\alpha_\mu}^{-1}(b_i)) \prod_{\mu \in s} (1 - F(g_{\alpha_\mu}^{-1}(b_i))) \right) db_i$$

$$= (1 - \alpha_i m) \sum_{s \in P_{m-1, N}^{-\{i\}}} \left(\prod_{j \notin s} F(g_{\alpha_j}^{-1}(b_i)) \prod_{j \in s} (1 - F(g_{\alpha_j}^{-1}(b_i))) \right)$$

Defining $\zeta_{\alpha_i}(x) = g_{\alpha_i}^{-1}(x)$, which means that $\zeta()$ are the inverse of $g()$, we derive the system of differential equations 32 for all possible values of $x = b_i$ and for every agent i with model α_i . The boundary condition is $g_{\alpha_i}(v_i^L) = v_i^L$, as in the case when the opponent models are not known. ■

Theorem 6 Consider the same setting as that of theorem 4, with the difference that the models α_i of all bidders are now common knowledge. Then, strategy $g_{\alpha_i}(v_i)$ constitutes a Bayes-Nash equilibrium, where $\zeta_{\alpha}(x) = g_{\alpha}^{-1}(x)$ is the solution of the system of differential equations:

$$\sum_{j=1, j \neq i}^N \zeta'_{\alpha_j}(x) F'(\zeta_{\alpha_j}(x)) (\zeta_{\alpha_i}(x) - x - \alpha_i \zeta_{\alpha_i}(x) + \alpha_i \zeta_{\alpha_j}(x)).$$

$$\sum_{s \in P_{m-1, N}^{-\{i, j\}}} \left(\prod_{\mu \notin s} F(\zeta_{\alpha_\mu}(x)) \prod_{\mu \in s} (1 - F(\zeta_{\alpha_\mu}(x))) \right) =$$

$$-\alpha_i m \sum_{s \in P_{m, N}^{-\{i\}}} \left(\prod_{j \notin s} F(\zeta_{\alpha_j}(x)) \prod_{j \in s} (1 - F(\zeta_{\alpha_j}(x))) \right) \quad (36)$$

with boundary conditions: $g_{\alpha_i}(v_i^H) = v_i^H, \forall \alpha_i$.

5 Conclusions

In this paper, we examined asymmetric bidder models both in risk attitudes, valuations and competitiveness. We gave the systems of differential equations that characterize the Bayes-Nash equilibria in these cases, both when the models of all bidders are common knowledge as well as when there is uncertainty about the models of the opponents. We examined both settings where m^{th} and $(m + 1)^{th}$ price auctions are being used.

There are still a number of issues we are currently pursuing. The foremost of these is that we are exploring different methods of solving the systems of differential equations which characterize the equilibria of this paper. In this paper, we did not present how to solve these equations, other than the example of section 3.1. Our research has concluded that these particular systems of differential equations are inherently unstable around the point specified by the boundary conditions. Therefore, when using methods based on the Taylor expansion, such as the Runge-Kutta variant we presented in [10], which can solve all these systems, it is not guaranteed that a solution will be found, even if one exists, exactly due to this instability. To this end, we are currently working also with symbolic solutions to these equations, which give the solutions of the systems presented in this paper, but are less general than the equivalent numerical methods. This will allow us to examine the cases when these solutions exist. Furthermore, it is clear that being able to analyze auctions with asymmetric bidders will facilitate the analysis of a number of real world scenarios; for example, we are examining the application of our results to service procuring scenarios.

Acknowledgements This research was undertaken under AL-ADDIN (joint EPSRC and BAE project EP/C548051/1).

REFERENCES

- [1] Felix Brandt, Tuomas Sandholm, and Yoav Shoham, 'Spitful bidding in sealed-bid auctions', in *IJCAI-07*, (2007).
- [2] Werner Güth, Radosveta Ivanova-Stenzel, and Elmar Wolfstetter, 'Bidding behavior in asymmetric auctions: An experimental study', *European Economic Review*, **49**, 1891–1913, (2005).
- [3] Vijay Krishna, *Auction theory*, Academic Press, 2002.
- [4] Bernard Lebrun, 'First price auctions and the asymmetric n bidder case', *Int. Econ. Rev.*, **40**(1), 125–142, (1999).
- [5] Y. Liu, R. Goodwin, and S. Koenig, 'Risk-averse auction agents', in *AAMAS-03*, pp. 353–360, (2003).
- [6] E.S. Maskin and J.G. Riley, 'Asymmetric auctions', *Review of Economic Studies*, **67**, 413–438, (2000).
- [7] William Press, Saul Teukolsky, William Vetterling, and Brian Flannery, *Numerical Recipes*, Cambridge University Press, 2007.
- [8] John A. Rice, *Mathematical Statistics and Data Analysis*, Duxbury Press, California, 1995.
- [9] I. A. Vetsikas and N. R. Jennings, 'Outperforming the competition in multi-unit sealed bid auctions', in *AAMAS-07*, pp. 702–709, (2007).
- [10] I. A. Vetsikas and N. R. Jennings, 'Considering asymmetric opponents in multi-unit sealed-bid auctions', in *Proceedings of the 1st International Workshop on Market Based Control (MBC)*, (2008).

Towards a Value-Sensitive System to support Agents in Norm Fulfillment and Enforcement

Sietse Overbeek, Virginia Dignum, and Yao-Hua Tan¹

Abstract. Increasing transparency in organizations has been an important driver for many changes in regulations in the last decade. Organizations that observe the regulations comply with the norms that are acceptable for the regulating body. These norms indicate desirable behaviors that should be carried out as well as undesirable behaviors that should be avoided. Norm enforcement mechanisms are used to determine whether organizations have complied with the norms, which can be divided into mechanisms that are oriented towards direct control and mechanisms that are oriented towards self regulation. Applying the mentioned norm enforcement mechanisms provides value for both the actors who require to fulfill norms and for the regulators who enforce norm fulfillment. When designing a system to support agents in norm fulfillment and enforcement, the values that are created by applying the norm enforcement mechanisms can be explicitly incorporated in the development of the system. In this paper, a first step in the development of such a value-sensitive system is taken by formalizing the values of direct control and self regulation. Finally, the remaining steps that are necessary to complete the development of the proposed value-sensitive system process are outlined. This provides a pathway to a full system implementation.

1 Introduction

For the last decade, organizations have been faced with an increase in regulation changes demanding transparency in an organization's books and its operational management. Examples of such changes are: the Sarbanes-Oxley (SOx) Act² intended for American organizations quoted on the stock exchange, the Dutch counterpart 'Tabaksblat'³, the Basel II Accords⁴ for banks, and the 'Markets in Financial Instruments Directive' (MiFID)⁵ for the European stock exchange. The introduction of these changes in regulation is the result of several business scandals, of which the scandals at *WorldCom*, *Enron*, *Parmalat*, *Satyam*, and *Ahold* are only a few of the many examples that stir the imagination (see e.g. [5]). To regulate organizations, the norms that organizations have to comply with play an important role [3]. A *norm* can be defined as standard behavior that is acceptable for the regulating body, indicating desirable behaviors that should be carried out as well as undesirable behaviors that should be avoided [10]. *Norm enforcement* mechanisms are used to determine

if organizations have complied to the *norms* that they should satisfy [4]. Such mechanisms are oriented towards *direct control* or towards *self regulation*. In direct control the regulating body or *regulator* for short directly controls the actions of *actors* in an organization. In this paper, an actor and a regulator are viewed as two roles that can be enacted by *agents*. An actor agent is a human or a computer-based entity that is able to carry out some task and a regulator agent is responsible for regulating some actor agent. In the regulating model of direct control, a regulator agent uses a *norm framework* to derive a set of norms tailored to an actor agent's specific situation. This implies that a norm framework contains a set of norms that need to be complied with by an actor for which the norm framework has been tailored. An actor is *sanctioned* if an actor does not comply with one or more norms that are part of a tailored norm framework. In the model of self regulation, actor agents are responsible to control their own behavior by using the norm frameworks that have been tailored to them. A self-regulating actor is sanctioned if a regulator determines that an actor fails to comply to the norm framework, despite its self-regulative activities.

Moral values are the standards of good and evil that guide an individual's behavior and choices [12]. Individuals, groups, and societies develop own value systems used for the purpose of ethical integrity. Successful norms and norm enforcement mechanisms are derived (or adhere to) such value systems. The value notion and the two mentioned different types of norm enforcement mechanisms can be combined to design a *value-sensitive system* that supports agents in *norm fulfillment* and *norm enforcement*. A first step is taken by explicating the value of direct control and self regulation by elaborating on the *formal values* of direct control and self regulation. These values are in fact the result of applying the *value interpretation* phase of a Value-sensitive System Development (VSD) process [1, 13].

The VSD process, which is depicted in figure 1, traces the influence of values in the design and engineering of systems. In particular, values are typically described in high level abstract ways (e.g., the natural language concepts of direct control and self regulation) that do not provide enough formality to be usable at the system specification level. Therefore, the values of direct control and self regulation are interpreted in natural language and translated to formal values in a formal language. The formal language used in this paper is first-order predicate logic [2, 6] combined with set theory [7, 8]. We have chosen for a first-order predicate calculus, because a second-order or a higher order theory admits already a part of the set theory in using its higher order variables [8]. Second-order variables are essentially set variables. Therefore, it does not seem right to consider some sets as first-order objects, while having around also second-order objects which are sets [8]. The translation to formal values will provide the basis for the remainder of the VSD process, eventually leading to a

¹ Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2600 GA Delft, The Netherlands, EU, email: {S.J.Overbeek,M.V.Dignum,Y.Tan}@tudelft.nl

² See e.g. http://en.wikipedia.org/wiki/Sarbanes-Oxley_Act

³ See e.g. <http://www.commissiecorporategovernance.nl/InformationinEnglish>

⁴ See e.g. <http://www.bis.org/publ/bcbsca.htm>

⁵ See e.g. http://en.wikipedia.org/wiki/Markets_in_Financial_Instruments_Directive

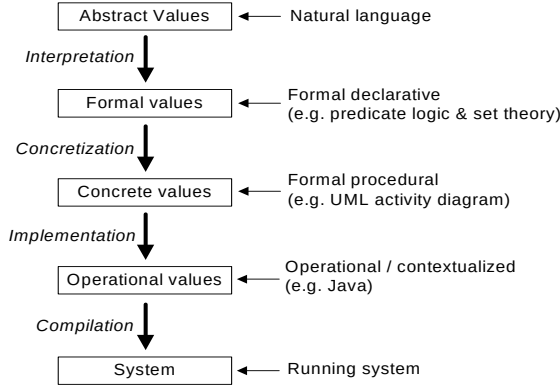


Figure 1. Value-sensitive system development process, based on [1, 13].

system that supports agents in direct control and self regulative contexts. This is shown in sections 2 and 3.

Steps towards the completion of the remaining phases of the VSD process, namely the design and engineering of a system to support agents, are outlined in section 4. As is shown in figure 1, the formal values can be further concretized by translating them in a formal procedural language. For example, the Unified Modeling Language (UML) includes an activity diagramming technique [11] that can be used to concretize formal values. More specifically, a UML activity diagram is used in section 4.1 to describe the operational step-by-step workflows of the direct control and self regulation mechanisms. The final step of the VSD process consists of translating the concrete values to operational values by implementing the concrete values by means of a programming language. In section 4.2, this implementation is outlined by showing an example Java code snippet. The conclusions of this paper and an overview of future research are presented in section 5.

2 Formalization of values

Following the VSD process, the formal values of *direct control* and *self regulation* can be realized by determining and understanding the corresponding abstract values. The norm enforcement mechanisms of direct control and self regulation have been elaborated in natural language in [4], which is used to interpret the abstract values. In direct control the regulator directly controls the tasks that are fulfilled by actors in an organization. In the model of self regulation, actors control their own behavior. A self-regulating actor is penalized if a regulator determines that an actor fails to comply to the norm framework, despite its self-regulative activities. The formal values are then based on these interpretations. Before the abstract values can be formalized, it is necessary to formalize the basics of direct control and self regulation that can be used to express formal values. One of those basics are the agent *roles* that are involved and have been explained in section 1. Formally, the set of agent roles can be denoted as $\mathcal{RO}$. In this case, the set contains the roles of *actor* and *regulator*:

$$\mathcal{RO} \triangleq \{\text{actor}, \text{regulator}\} \quad (1)$$

A role enactment is a specific *fulfillment* of such a role by an agent, expressed by the function:

$$\text{Enact} : \mathcal{RE} \rightarrow \mathcal{RO} \quad (2)$$

The set $\mathcal{RE}$ is the set of all role enactments within the model of direct control. Given the role enactment e of a role $\text{Enact}(e)$, we can view the actor that specifically enacts the role as a function:

$$\text{Player} : \mathcal{RE} \rightarrow \mathcal{AS} \quad (3)$$

The set $\mathcal{AS}$ is the set of agents. Since an enactment indicates an actor ‘in a role’ we know that an actor and a role combination uniquely determines an enactment:

$$\text{Player}(e_1) = \text{Player}(e_2) \wedge \text{Enact}(e_1) = \text{Enact}(e_2) \Rightarrow e_1 = e_2 \quad (4)$$

For an enactment $e \in \mathcal{RE}$ the following notation is introduced:

$$_ \rightsquigarrow _ \subseteq \mathcal{AS} \times \mathcal{RE} \times \mathcal{RO} \quad (5)$$

$$a \rightsquigarrow^e r \triangleq \text{Player}(e) = a \wedge \text{Enact}(e) = r \quad (6)$$

$$a \rightsquigarrow r \triangleq \exists e \in \mathcal{RE} [a \rightsquigarrow^e r] \quad (7)$$

This can be illustrated by the following *running example*. Let a tax officer denoted as a be an agent that can play the two mentioned roles. He either plays the role of type *actor* denoted by r_1 , or the role of type *regulator* denoted by r_2 . Suppose that a tax officer working at the national Tax Administration inspects the completed tax returns for a citizen. In such a situation, the tax officer enacts the role of regulator, because he checks whether the tax returns are excluded from any tax violations. In a different evaluative situation, for example a job performance evaluation, the manager of the tax officer may inspect whether the officer is functioning properly. In that case, the tax officer enacts the role of *actor* who is being regulated. Combining the example with the aforementioned formalisms, it can be stated that both $a \rightsquigarrow^{e_1} r_1$ and $a \rightsquigarrow^{e_2} r_2$ are enactments such that $\text{Player}(e_1) = a$, $\text{Player}(e_2) = a$, $\text{Enact}(e_1) = r_1$, and $\text{Enact}(e_2) = r_2$. Finally, a set of agents all enacting a certain role can be defined as follows:

$$\mathcal{AS}_r \triangleq \{a \in \mathcal{AS} | a \rightsquigarrow r\} \quad (8)$$

It can now be said that the set $\mathcal{AS}_r$ includes the agents that play role r in the models of direct control or self regulation. If agent x enacts the actor role in the model this can be denoted as follows: $x \in \mathcal{AS}_{\text{actor}}$.

A task *instance* that is fulfilled by a specific agent can be viewed as a function:

$$\text{Fulfillment} : \mathcal{TI} \rightarrow \mathcal{AS} \quad (9)$$

The set $\mathcal{TI}$ is the set of task instances that are fulfilled by an agent. A task instance $i \in \mathcal{TI}$ that is fulfilled by an agent $a \in \mathcal{AS}$ can be expressed as: $\text{Fulfillment}(i) = a$. A specific instantiation of a task *type* is expressed as:

$$\text{TaskInst} : \mathcal{TI} \rightarrow \mathcal{TT} \quad (10)$$

The set $\mathcal{TT}$ is the set of *task types* that can be instantiated by a specific task instance. The expression $\text{TaskInst}(i) = t$ can be used to assert that a task instance i is characterized as a task of the type t . Task fulfillment can be illustrated by the tax officer who inspects the tax returns. In this case, the inspection of the tax returns is a task instance that is fulfilled by the tax officer. Such a task instance can then be classified as a task of the type ‘tax return inspection’ for example.

Such as is stated in section 1, a regulator agent uses a *norm framework* $n \in \mathcal{NF}$ to regulate an actor agent, where $\mathcal{NF}$ is the set of norm frameworks. Some actor agent $b \in \mathcal{AS}$ complies with the norms that are applicable to agent b . The set of norms is denoted as $\mathcal{NS}$. An actor agent can violate the norms by pursuing an illegal goal or by performing an illegitimate action, i.e. when an actor agent does not comply to the norm framework. In direct control, an actor will get

penalized by a regulator. In self regulation, an actor will try to prevent this kind of unwanted behavior beforehand. It is assumed that a regulator agent uses the norm framework to derive a set of norms tailored to an actor's specific situation. The norms that are contained in a norm framework can be formalized as follows:

$$\text{Containment} : \mathcal{NF} \rightarrow \wp(\mathcal{NS}) \quad (11)$$

If a norm framework n contains a set of norms $S \subseteq \mathcal{NS}$, this can be expressed as: $\text{Containment}(n) = S$. For example, if a tax officer detects that a tax return does not comply with one or more norms the officer can penalize the responsible agent by imposing a fine. An example of a norm that must be fulfilled when completing a tax return is the norm 'provide an overview of all collected earnings'. The actor agent will risk a fine if this norm is not complied with and if the tax officer detects the failure to comply with this norm. An agent that uses some norm framework to regulate another agent can be found by using the regulator function, while an agent that is regulated by another agent can be found by using the regulatee function:

$$\text{Regulator, Regulatee} : \mathcal{NF} \rightarrow \mathcal{AS} \quad (12)$$

Both functions are used to understand which *regulators* regulate *actors* using some *norm framework*:

$$- \triangleq - \subseteq \mathcal{AS} \times \mathcal{NF} \times \mathcal{AS} \quad (13)$$

$$a \triangleq^n b \triangleq \text{Regulator}(n) = a \wedge \text{Regulatee}(n) = b \quad (14)$$

$$a \triangleq b \triangleq \exists n \in \mathcal{NF} [a \triangleq^n b] \quad (15)$$

The notation $a \triangleq b$ 'expresses' that agent a regulates agent b using some norm framework n . In the tax return example, it can be said that the tax officer regulates the citizen using a norm framework that applies when completing a national tax return form. Such a norm framework is issued by the national Tax Administration. Next, the comply function can be used to indicate whether an actor agent does comply with a set of norms included in a norm framework or not. This function is formalized as follows:

$$\text{Comply} : \mathcal{AS} \rightarrow \wp(\mathcal{NS}) \quad (16)$$

The sanctioning function, which is almost equal to the comply function, is introduced to penalize an actor agent if he has not complied with the norms:

$$\text{Sanctioning} : \mathcal{AS} \times \mathcal{AS} \rightarrow \wp(\mathcal{SN}) \quad (17)$$

The set $\mathcal{SN}$ is the set of possible sanctions that can be imposed on an actor. At this point, the fulfillment function, the notation to express regulation, the containment function, the comply function, and the sanctioning function can be combined to express that:

- An agent $x \in \mathcal{AS}$ regulates an agent $y \in \mathcal{AS}$ by using a norm framework $n \in \mathcal{NF}$.
- Agent $x \in \mathcal{AS}$ must be a regulator and agent $y \in \mathcal{AS}$ must be an actor.
- Actor agent y fulfills a task.
- An actor agent y complies or fails to comply with one or more of the norms contained in a norm framework $n \in \mathcal{NF}$ that is used by a regulator.
- In case an actor agent y fails to comply, he will get penalized by regulator x in terms of one or more sanctions contained in the set $S \subseteq \mathcal{SN}$.

At this point, the *abstract values* of *direct control* and *self regulation* can be applied by using the introduced formalisms.

3 Applying formal values

At first, the formal interpretation of direct control is expressed by means of equations 18 and 19.

3.1 The formal value of direct control

The case in which an actor indeed complies with all norms during the fulfillment of a task can be expressed as follows:

$$\begin{aligned} \exists i \in \mathcal{TI} \exists x, y \in \mathcal{AS} \exists n \in \mathcal{NF} [x \triangleq y \wedge \text{Fulfillment}(y) = i \wedge \\ \text{Containment}(n) = \text{Comply}(y) \wedge \\ x \rightsquigarrow \text{regulator} \wedge y \rightsquigarrow \text{actor}] \end{aligned} \quad (18)$$

The tax return example can be used to further explain this equation. Assume that a tax officer x regulates a citizen y by inspecting a citizen's completed tax form and that no flaws are detected. This means that the norms n that are contained in some norm framework equate to the norms that actor y complies with during the fulfillment of a task i in which a citizen y completes a tax form.

Unfortunately, it is imaginable that an actor does not comply with one or more norms included in the norm framework that is used to regulate such an actor. Because of that, actor y will get penalized by one or more sanctions $S \subseteq \mathcal{SN}$ that are imposed on him. This can be expressed as follows:

$$\begin{aligned} \exists i \in \mathcal{TI} \exists x, y \in \mathcal{AS} \exists n \in \mathcal{NF} \exists S \subseteq \mathcal{SN} [x \triangleq y \wedge \text{Fulfillment}(y) = i \wedge \\ \text{Containment}(n) \neq \text{Comply}(y) \wedge \text{Sanctioning}(x, y) = S \wedge \\ x \rightsquigarrow \text{regulator} \wedge y \rightsquigarrow \text{actor}] \end{aligned} \quad (19)$$

This formalism can be further explained by embroidering on the tax return example. One or more sanctions S are issued by the tax officer in case the norms n that are contained in some norm framework do not equate to the norms that actor y should have complied with. A sanction can be a fine or the reclaim of unlawfully obtained tax money. In the formal model of direct control, a regulating agent directly controls the actions of an actor agent. In the formal model of self regulation, which is discussed next, the actor agent regulates itself.

3.2 The formal value of self regulation

The formalisms introduced in the previous section can be reused to conceptualize the formal value of self regulation. The case in which an actor complies with all norms during the fulfillment of a task is expressed in a different way in the model of self regulation when compared to the model of direct control:

$$\begin{aligned} \exists i \in \mathcal{TI} \exists x \in \mathcal{AS} \exists n \in \mathcal{NF} [x \triangleq x \wedge \text{Fulfillment}(x) = i \wedge \\ \text{Containment}(n) = \text{Comply}(x) \wedge x \rightsquigarrow \text{actor}] \end{aligned} \quad (20)$$

There is no agent involved that enacts the role of regulator if an actor complies with all the norms contained in his norm framework. This implies that an actor agent can regulate himself without the involvement of a regulator agent.

In the model of self regulation, a regulator only appears in case an actor does not comply with one or more norms included in the norm framework. This happens if a regulator establishes that the set of norms contained in a norm framework is not equal to the set of

complied norms. This observation is followed by imposing one or more sanctions on the actor agent. This can be formalized as follows:

$$\begin{aligned} \exists i \in \mathcal{IT} \exists x, y \in \mathcal{AS} \exists n \in \mathcal{NF} \exists s \in \mathcal{SN} [x \simeq x \wedge \text{Fulfillment}(x) = i \wedge \\ \text{Containment}(n) \neq \text{Comply}(x) \wedge \text{Sanctioning}(y, x) = S \wedge \\ y \rightsquigarrow \text{regulator} \wedge x \rightsquigarrow \text{actor}] \quad (21) \end{aligned}$$

The formal values of direct control and self regulation have been specified up till now. Translated to the VSD process, this implies that an interpretation of abstract values has been made resulting in formal values that have been specified in a formal declarative language. Next, an outline of the remaining system development phases is presented to get an impression of what is needed to develop a value-sensitive system that is able to support agents in norm fulfillment and enforcement.

4 From formal values to a running system

The formal values that have been introduced are expressive enough to enable a discussion with stakeholders of a value-sensitive system that has to be developed, but do not provide enough information on how to build a system that complies with those values. Formalizing the values will offer precise syntax and semantics preventing ambiguity, provided that a discussion facilitator is available who understands these formalisms in case none of the stakeholders can interpret them. *Concrete value* descriptions, which are the result of further concretizing the formal value descriptions, specify the behavior of the system, define constraints, and indicate how to react in case of unwanted behavior. In fact, developing concrete value descriptions corresponds to general analysis and design steps in system development. Examples of system design artifacts reflecting concrete values are Unified Modeling Language (UML) diagrams such as an activity diagram, a sequence diagram, and a collaboration diagram. Other examples include a Business Process Modeling Notation (BPMN) model, a Petri net, and a Data Flow Diagram (DFD). As part of the outline of the remaining development phases of a value-sensitive system, the formal values of direct control and self regulation are concretized by designing two activity diagrams.

4.1 Concrete values

The formal value of direct control expressed by means of equations 18 and 19 can be concretized as a UML activity diagram shown in figure 2. Activity diagrams can be divided into object swimlanes that determine which object is responsible for which activity [11]. The direct control activity diagram shows the swimlanes for an actor and a regulator. Arrows indicate transitions coming out of each activity, connecting it to the next activity. A transition may branch into two or more mutually exclusive transitions. Guard expressions shown inside brackets label the transitions coming out of a branch. A branch appears in the diagram as a hollow diamond. The activities that are performed after the ‘control task’ activity which is carried out by a regulator are dependent of the outcome of an actor’s norm fulfillment performance. Sanctions will follow in case the regulator finds out that an actor does not comply to one or more norms. After dealing with those sanctions, an actor can finish working if there are no tasks left or he can continue with another task. A regulator can finish controlling if an actor has ceased working or he can continue to control if an actor is going to fulfill remaining tasks.

The formal value of self regulation expressed by means of equations 20 and 21 can be concretized as a UML activity diagram shown

in figure 3. When comparing both activity diagrams it is clear that the ‘control actor’ activity is not a regulator’s responsibility in the case of self regulation. The only two activities that are left for a regulator are the tailoring of a norm framework to an actor and the issuing of sanctions when norms are not fulfilled. Figure 1 shows that the final step of the VSD process consists of the implementation of concrete values to get operational values. Operational values are the codification of functions in a system level language that contributes to the value implementation. An illustration of a Java code snippet is shown in the next section that operationalizes the ‘interpret norm framework’ activity of the actor to further outline the value-sensitive system.

4.2 Operational values

The activity ‘interpret norm framework’ can be found in both activity diagrams and is performed by an actor. It would go beyond the scope of this paper to completely operationalize the concrete values, but a code snippet will partly illustrate the operationalization of those values. Figure 4 shows the ‘NormFramework’ class, in which the contents of the ‘normframework.txt’ file containing attribute-value pairs is read, and each line is parsed leading to a list of generated norms. An attribute-value pair is a fundamental data representation, in which

```
public class NormFramework {
    private final File file;

    public static void main(String args[]) throws FileNotFoundException {
        NormFramework parser = new NormFramework("\\server\\normframework.txt");
        parser.processFramework();
    }

    public NormFramework(String filename) {
        file = new File(filename);
    }

    private void processFramework() throws FileNotFoundException {
        Scanner scanner = new Scanner(file);
        try {
            while (scanner.hasNextLine()) {
                processNorm(scanner.nextLine());
            }
        } finally {
            scanner.close();
        }
    }

    private void processNorm(String line) {
        Scanner scanner = new Scanner(line);
        scanner.useDelimiter(",");
        if (scanner.hasNext()) {
            String name = scanner.next();
            String value = scanner.next();
            System.out.println("Norm type: " + name.trim());
            System.out.println("Norm instance: " + value.trim());
            System.out.println();
        } else {
            System.out.println("Process failure.");
        }
        scanner.close();
    }
}
```

Figure 4. Source code snippet of the ‘NormFramework’ class.

all or part of the data model may be expressed as a collection of tuples <attribute, value>, see e.g. [9]. Table 1 shows four example attribute-value pairs that are part of the tax return example. These

Table 1. Example attribute-value pairs and parsed output.

Norm type	Norm instance
Venture capital	The maximum amount for deductible venture capital < €46.984
Income	The tax free income is €20.661
Deduction	Senior deduction is granted if age > 64 and collective income < €34.282
Deduction	Bonus for work continuation is granted if birthdate < 1948 or if work income > €8.859
...	...

pairs are included in the ‘normframework.txt’ text file that is parsed to generate norms that apply for citizens who complete tax forms.

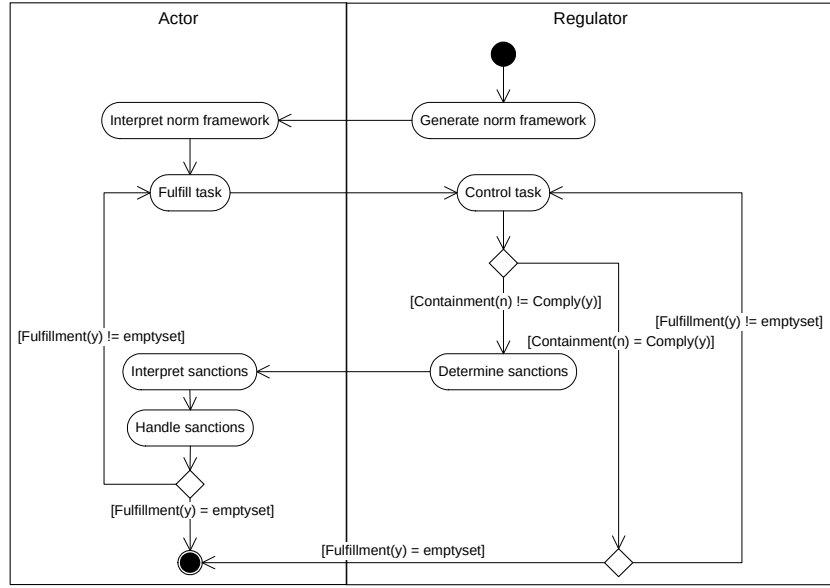


Figure 2. Activity diagram reflecting the concrete value of direct control.

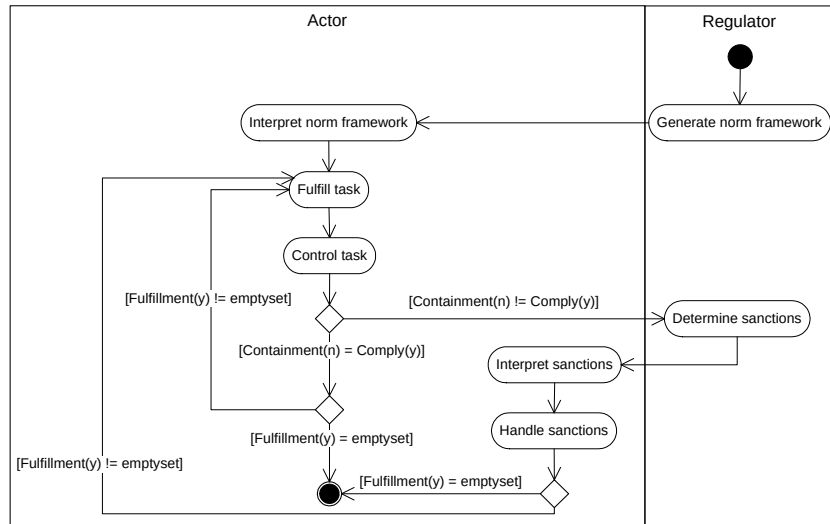


Figure 3. Activity diagram reflecting the concrete value of self regulation.

In other words, the text file represents a norm framework that is tailored to citizens who complete tax forms. Table 1 shows norms of a certain type (the attributes) with their corresponding instances (the values of the attributes). The norm types that are shown are related to venture capital, income, and deduction and should be taken into account when completing tax returns. Specific instances of these types are shown in the right column of the table. Once the norm framework is parsed, it can be interpreted by an actor to understand which norms have to be complied with when fulfilling tasks. Subsequently, the next activity shown in the activity diagrams can be performed. Finally, an overview of the current static structure of the value-sensitive system to support agents in norm fulfillment and enforcement is pro-

vided.

4.3 Static structure diagram

Figure 5 shows a UML class diagram that shows all classes and their relationships of the outlined value-sensitive system to map out the system's structure. The equations that have been introduced in section 2 and 3, the activities as part of the activity diagrams, and the source code of the norm framework class have been used to design the class diagram. The *sets* that are part of the formal values are the *classes* in the class diagram. The relationships between the classes can be identified by studying the equations that constitute the formal

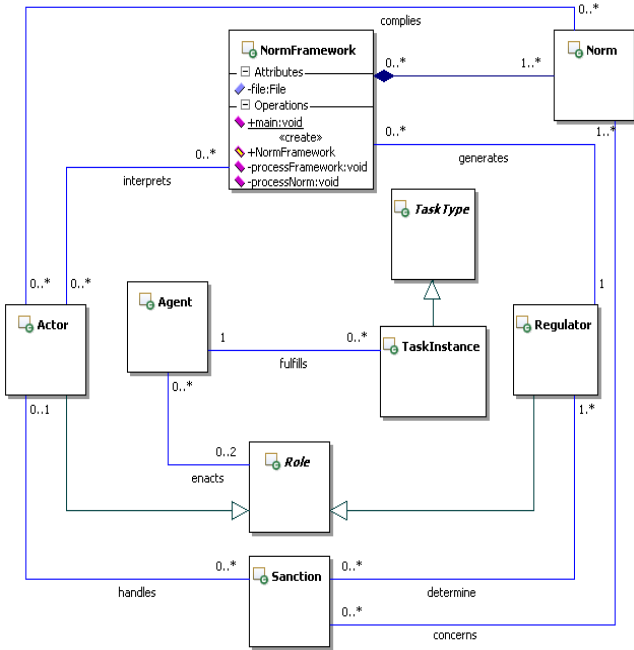


Figure 5. Class diagram reflecting the static structure of the system.

values and the activities in the activity diagrams. The norm framework class shows the ‘file’ attribute and the operations that are part of the source code of this class shown in figure 4. The *composition* relationship between the norm framework class and the norm class shows that every norm framework is composed of at least one or more norms. Furthermore, three generalization relationships are shown. Two of the generalization relationships show that the actor class and the regulator class are subtypes of the *abstract* role class. The role class is abstract to indicate that the class itself cannot be instantiated, but its child classes ‘actor’ and ‘regulator’ can be instantiated instead. The third generalization relationship shows that the *abstract* task type class is a supertype of the task instance class. The generalization shows that a task instance is of a certain type, e.g. all task instances that are related to tax return inspections can be classified as tasks of the type ‘tax return inspection’. The task type class is abstract, because agents fulfill a task *instance* instead of a *type*. The class diagram provides a route to future research in which attributes and operations can be added to the classes. Furthermore, this exercise will then pave the road for a full system implementation.

5 Conclusions & future research

The results of the presented research provide the basis for a value-sensitive system to support actor agents in norm fulfillment and regulating agents in norm enforcement. This foundation has been laid by applying a value-sensitive system development process and by incorporating the principles of the norm enforcement mechanisms of direct control and self regulation in the system design. By following this specific system development process, the value that is created for the agents that apply the norm enforcement mechanisms of direct control and self regulation is explicitly incorporated in the development of the system. The process consists of three phases: value interpretation, value concretization, and implementation of the val-

ues into the development of the system. The interpretation phase has been fully described by means of formalisms that express the values of direct control and self regulation in an exact and precise manner. The remaining phases have been outlined by providing two UML activity diagrams that describe the workflows of the direct control and self regulation mechanisms and by implementing one of system classes. Finally, the static structure of the current system design has been mapped out in a UML class diagram.

Future research is aimed at developing a simulation of a situation in which agents are regulated by means of direct control and a situation in which agents are regulated by means of self regulation. This simulation will be used to evaluate, improve and extend the models that have been presented in this paper. Besides that, it will further increase our understanding of the similarities and differences between these norm enforcement mechanisms as well as the advantages and disadvantages of each mechanism.

ACKNOWLEDGEMENTS

This work was supported by the ABC project (acronym for Agent-mediated Business Coordination). ABC is funded by the Netherlands Organisation for Scientific Research (NWO) under grant number 612.066.515.

REFERENCES

- [1] H. Aldewereld, *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*, Ph.D. dissertation, Utrecht University, The Netherlands, EU, 2007.
- [2] J.F.A.K. Benthem, H.P. Ditmarsch, J. Ketting, J.S. Lodder, and W.P.M. Meyer-Viol, *Logica voor Informatica*, Pearson Education Benelux, Amsterdam, The Netherlands, EU, third edn., 2003. In Dutch.
- [3] G. Boella, L. Lesmo, and R. Damiano, ‘On the ontological status of plans and norms’, *Artificial Intelligence and Law*, **12**(4), 317–357, (2004).
- [4] B. Burgemeestre, J. Hulsteijn, and Y.-H. Tan, ‘Towards an architecture for self-regulating agents: a case study in international trade’, in *Proceedings of the Second Multi-Agent Logics, Languages, and Organizations Federated Workshops*, Turin, Italy, September 7-10, 2009, volume 494 of *CEUR Workshop Proceedings*. Sun SITE Central Europe, Aachen, Germany, EU, (2009).
- [5] R.M. Grant and M. Visconti, ‘The strategic background to corporate accounting scandals’, *Long Range Planning*, **39**(4), 361–383, (2006).
- [6] A.G. Hamilton, *Logic for Mathematicians*, Cambridge University Press, Cambridge, UK, EU, 1978.
- [7] A.H.M. ter Hofstede and H.A. (Erik) Proper, ‘How to formalize it?’, *Information and Software Technology*, **40**(10), 519–540, (1998).
- [8] A. Levy, *Basic Set Theory*, Dover Publications, Mineola, NY, USA, 2002.
- [9] M. Magnani and D. Montesi, ‘A unified approach to structured and XML data modeling and manipulation’, *Data & Knowledge Engineering*, **59**(1), 25–62, (2006).
- [10] F.R. Meneguzzi and M. Luck, ‘Norm-based behaviour modification in bdi agents’, in *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Budapest, Hungary, eds., C. Sierra, C. Castelfranchi, K.S. Decker, and J.S. Sichman, volume 1, pp. 177–184. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, USA, (2009).
- [11] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, The Addison-Wesley Object Technology Series, Boston, MA, USA, second edn., 2004.
- [12] E.D. Scott, ‘Organizational moral values’, *Business Ethics Quarterly*, **12**(1), 33–55, (2002).
- [13] M.J. van den Hoven, ‘Design for values and values for design’, *Information Age +, Journal of the Australian Computer Society*, **7**(2), 4–7, (2005).

An Agent Based Recommendation System for Tourism

João Laranjeira^{1,3}, João Carneiro^{1,3}, Goreti Marreiros^{1,3}, Ricardo Santos^{1,2}, Carlos Ramos^{1,3}

Abstract. Tourism might be seen as an activity in a different place from the one we are living in. Considering the majority of the population lifestyle, time factor is one of the most main reasons to restrain practice of different activities. Not only for the time that is necessary to accomplish the activity but also for the time spent on planning and organizing it. Alongside with time there are other factors that prevent people from enjoying such moments. For an activity planning it is necessary time for an individual to think on its possible activities. Knowing all these factors, we can easily understand that a Recommendation System will be an extremely useful tool being able to help tourist in decisions, as well as in activities planning. The Recommendation System has information about the tourists and its possibilities to analyse it in order to understand the best way to satisfy the tourist's interests. The conclusions obtained by the Recommendation System will be suggested to the tourist.

The aim of this paper is to show how a Recommendation System can assist tourists in their choices, not only the ones based on the usual variables such as time, money, among others, but also on their interests. Furthermore, we turned to agents that simulate tourist to enrich the system with the typical social aspects of human.

To be able to simulate the tourists in a virtual world, allowing them to communicate with each other, exchanging information and evolving its profile and knowledge, the presented recommendation system will be based on an architecture based on agents. For each agent that reflects a tourist has anthropomorphic characteristics, increasing their capacity for flexibility and evolution over time.

1 INTRODUCTION

When talking about an area such as tourism it is necessary to take into consideration a broad range of factors in order to understand how these determine the motivations of tourists. Motivation is an important variable on the consumer and it is stimulated by a complex mix of economic, social, psychological, cultural and political influences related to industry and the wider environment in which the tourists are involved [1].

Moutinho [2] argued that motivation is a state of necessity, a condition that puts pressure on the individuals to certain types of activities that bring satisfaction. Thus, only realizing the true motivations of tourists we can understand what they really want, in order to present the best solutions that even they would never have taken into account.

Today, besides traditional travel agents, there are several Recommendation Systems to support tourists in the decision making process.

On the one hand, travel agents as humans have limitations on the amount of information they can handle comparing to a Recommendation System, on the other hand Recommendation System only help tourists based on an analysis of a limited set of attributes (for example, available time and money).

Nowadays, most people when plan a trip firstly will perform a Internet searching. More and more people take advantages of new technologies to plan their leisure activities [3]. Nevertheless, it is known that for many it is extremely difficult to obtain quality information [4].

Another reason that explains the difficulties of this process is that besides their own interests and preferences, users often find the need to take into account other people (for example, when families travel together) [5].

Usually, after reaching the desired location, tourists want to know the different activities available, in order to choose what they will do. Obviously it would be much better if tourists were able to decide about these anywhere and anytime. In our approach to the concept of a Recommendation System applied to tourism, it should be possible to suggest possible destinations, activities or points of interest to the tourist.

The system that is presented will try to represent the tourists' profiles as well as acquire knowledge about their activities through time.

Agent Based Recommendation System is considered an important tool in a broad range of areas e.g. individual decision making, e-commerce, traffic simulation, entertainment [6]. We claim that agent based recommendation system can be used with success in our presented architecture.

In our multi-agent system will exist two main types of agents: Agent Adviser, who will be responsible for the Recommendation Module and the Agent Tourist(an agent as this will represent the tourist in this virtual world).

In order to facilitate understanding of the topics that are addressed in this article we consider a scenario to explain the techniques presented.

Let us consider for the scenario, a tourist male gender, 35 years old and with the following preferences:

¹GECAD – Knowledge Engineering and Decision Support Group, Portugal, {jpcl, jmrc}@isep.ipp.pt, {goreti, csr}@dei.isep.ipp.pt, rjs@estgf.ipp.pt

²School of Management and Technology of Felgueiras – Polytechnic of Porto, Portugal

³Institute of Engineering – Polytechnic of Porto, Portugal

independence, personal goals, creativity, intellectual growth and companionship needs.

2 TOURIST PROFILE

Nowadays the biggest problem in Recommendation Systems of tourism is precisely the assessment of tourist in psychological terms. Their age and sex, as well as their tastes are varied which complicates the process in question. So it will be necessary to place every tourist in a certain group, with the objective of complete their profile processing in a more scientific and precise way, facilitating the process of decision making in tourism.

Tourists, like all humans, are influenced directly or indirectly by various factors such as lifestyle, the phase life cycle and gender. Gibson and Yiannakis[7] conducted a study that proved that tourists will be divided by their age, gender and preferences. So we can fit the tourist in a group and qualifies him as to the type of tourist [7]. This approach allows reducing the potential types of tourists and qualifying them relatively to their type with more assurance.

In order to pass the theoretical model [7] for the computational model it will be used a decision tree with the following variables: gender, age and preferences. The decision tree is used in the first time the user (tourist) accesses the system because the classification of tourist will be done only once. There are other mechanisms to classify tourist throughout his working life in the system, which will be explained later.

Figure 1 represents the decision tree that will be used to classify tourists according to its type, for the scenario introduced in this paper.

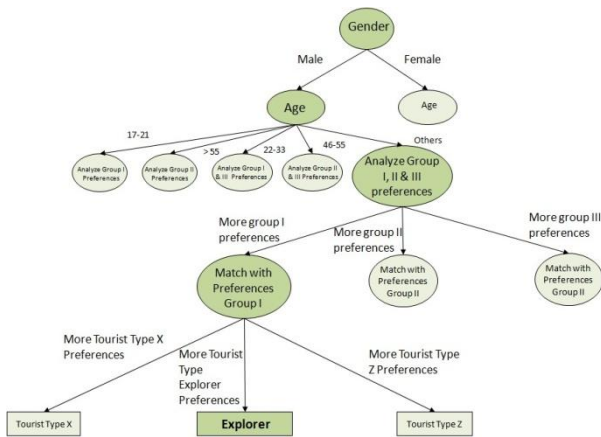


Figure 1 - Decision Tree for Scenario

This tree is divided in two parts of analysis. The first analysis, will take into consideration the age and sex of the tourist, being assigned to a group. In the second step after the assigned group, is made the match of the preferences related to the tourists, with all pre-defined preferences for the types of tourist standard, which are in the group which belongs to the tourist. In the end, the type of the tourist that has more similarities with the preferences of defined by the user will then be chosen to identify the tourist.

As it is possible to see in figure 1, the decision tree blamed the Explorer type to analyzed tourist. The path taken by the decision tree is identified by the color green.

3 TOURIST RECOMMENDATION

The recommendation to tourists is a crucial part in a recommendation system in tourism because this is a core functionality that a tourist has a tendency to look for a system with these characteristics. In order to satisfy the tourist as much as possible, the presented system is composed of two types of mechanisms recommendation. A mechanism will take into consideration the preferences of tourists and preferences associated with cities, activities or points of interest existing in the database, while the other mechanism will take into account the cities visited by tourists with identical preferences.

3.1 Utility Function

Table 2 describes a test case for a recommendation scenario introduced. Through the decision tree was assigned the type of tourist to the tourist Explorer analyzed, so this test case will be aimed at a tourist-type Explorer and male gender.

To obtains results we used an algorithm with an utility function to solve multi-criteria problems. The function used is the following:

$$f(r,p) = \sum_{i=1}^n \frac{r_i \times p}{n} \quad (1)$$

Where r is the rating of the preference of certain city and p is the weight of this city.

The process stems from preferences associated with the tourist preferences related to a particular city in the database. Possible cities are obtained after all and town is calculated by the score of all common preferences among tourists and the city.

Table 1. Cities Recommendations

City	Score
Lisbon	0.045
Brasilia	0.090
Madrid	0.180
Kabul	0.045
Luanda	0.135
Brussels	0.225
Habana	0.090
Athens	0.090
Roma	0.270
Amsterdam	0.225
London	0.270

If the objective were to pick the five best cities, then the system would recommend the cities of London, Rome, Amsterdam, Brussels and Madrid, once those are the cities with the highest score.

Activities that tourists want to take or points of interest that tourists want to visit are also obtained through the utility function. The process performed is identical to the process described earlier.

3.2 Case-Base Reasoning

In figure 2 we can see the recommendation process to the tourist cities into consideration the history of the trips made by tourists with the same type as the tourist who will be advised. Cities recommended are those cities that were evaluated further by tourists with same type that visited this city. So we can advise tourists based on the certainty that the tourists of same type liked to visit this city, avoid that recommendation of cities visited were evaluated with negative points.

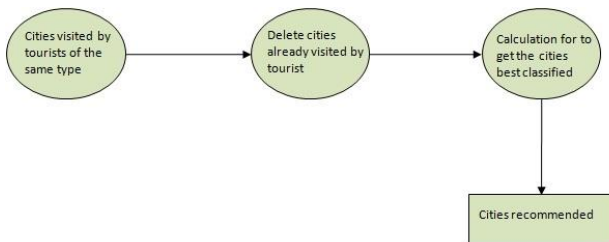


Figure 2 – Recommendation process based in the history of the tourist

In own scenario the tourist is an Explorer type, right to the use of case-based reasoning will be analyzed all the cities that were visited by tourists with the type Explorer. The recommended cities will be those who have better overall classification.

4 PRESENTED ARCHITECTURE

In order to put into practice the concepts it will be presented a presented architecture that will be the basis of the presented system.

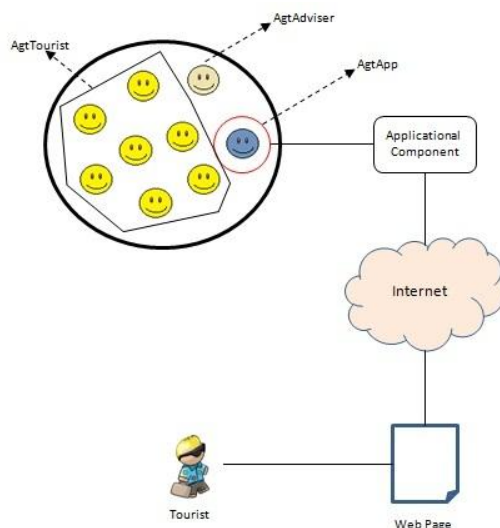


Figure 3 – Presented Architecture

In the previous figure we can see the presented architecture to support the system.

To enable user interaction with the system, it consists of a web page, making it a ubiquitous system, on the page are all features to collect and display information typical of recommendation systems of this type of business. In the

core of the architecture is the community of agents. In community of agents are agents who represent the tourists, the agent adviser and the agent applicational. The agent applicational acts as intermediary between the community of agents and applicational layer. The other two types of agents will be explained later.

The applicational component includes the analysis of tourist profile in order to classify the tourist on by style. After the visitor to register on the web page, data are sent to the applicational component. This component defines the profile of the tourist and is sent to the agent applicational the request to create an agent-type tourist, modeled with the profile set.

This architecture will allow the information to evolve naturally and the actions being taken by tourists to influence the entire community of agents. Thus, importance is given to all actions committed by tourists, both the navigation on the page and reactions to the recommendations that are being made, as the journeys and ratings. It is intended that the data can go through the whole system and that each agent can learn from the world around him, especially with tourists from agents representing similar profiles.

4.1 Agent Tourist

Each tourist agent is the responsible for representing a tourist in this virtual world. Thus, each agent is modeled to the personality of the tourist it represents. The agent has all the information relating to trips made by tourists: the places visited, the activities performed, points of interest visited, the score given by the tourist to a particular travel and evolutionary module. The evolutionary module which has been incorporated into the tourist agent is composed by two components: the development of personality and consulting travel destinations. The evolution of personality is a component that enables the agent to evaluate the trends of other agents with the same personality. The evaluation of these trends is related to decisions taken by tourists in their choice of destinations. This system features allows a user which has not accessing the system over a long period of time may eventually be recommended efficiently. This is possible because the agent profile is constantly updated with the interests of other tourists. The consultation component of travel destinations, allows the agent to collect information from other agents that have similar characteristics. This information is relevant to the travel that tourists have made and what the ratings assigned to them have to travel. Using these components there is a sharing of information as if it were a social network. This feature significantly increases the chance of a holiday destination to be advised that it will actually please the tourist in all aspects of this value.

4.2 Agent Adviser

The agent adviser is responsible for suggesting possible cities, activities or points of interest to be visited by tourists. To make recommendations this agent uses the

techniques previously described: the utility function and the case-based reasoning. Using a multi-agent system also aims that the agent adviser can do a quick search to respond to certain requests made by tourists on the web page. Thus this agent may communicate with the agent representing the tourist who is currently using the website, so that it can store information on the reactions taken by tourists on certain proposals, obtaining recommendations with increasing quality.

5 REASONING AGENTS

In this section we will explain the various processes that are added to the agents who compose the community of agents.

5.1 Creating Agents

The process of creating an agent begins when visitors makes the registration on the website and answer the questionnaire that creates an initial profile. The data is sent to the applicational component, where through the use of the decision tree (section 2) is possible to identify the profile of the tourist in question. After processing the data into useful information and identifiable by the system, that information is sent to the AgtApp, which gives order to create a new agent, modeled on the acquired information. In figure 4 you can see the flow of existing data, provided that the tourist is recorded on the website until the agent who will reflect this virtual world is created. The interaction that occurs between agents is done through the ICL (Interagent Communication Language), which is the language used for communication between agents in the OAA (Open Agent Architecture).

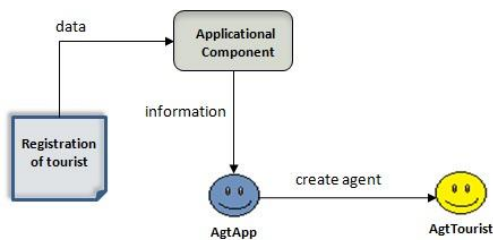


Figure 4 – Creating Agent Process

5.2 Agent Adviser Process

A major objective of our Recommendation System is to allow the access by the agent tourist to the relevant information to him in the system, thus being easier to learn from each tourist agent.

As in reality, actions taken by people affect the world around them. In this proposal, the aim is also that actions made by tourists affect the community of agents, i.e., affect each of the agents that reflect all the other tourists.

Whenever a tourist returning from a trip and he make login in the system, he have the possibility to classify the trip held at a given scale. When the Agent Adviser has a relevant set of new classifications within the group of tourists from a particular profile, it throws a notice to the community of agents, warning that there is new

information for a particular profile. This action will allow that the agents of the same profile are aware of the last trips made by tourists with its profile and the classification assigned to them. The way in which agents gain access to that knowledge and learn from it, is explained in the next section.

The following figure reflects the flow of information and the various states that it takes, also demonstrating how the rating of a trip made by a tourist, influences the community of agents.

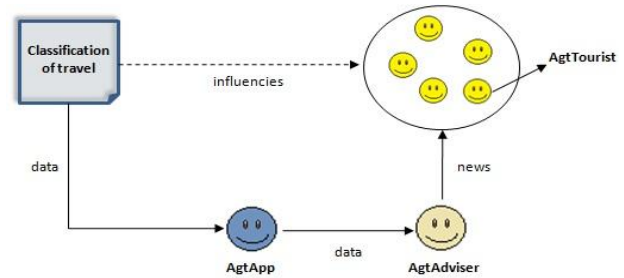


Figure 5 –Influence of Community Process

5.3 Communication Process between Agents Tourists

As stated previously, the agents' tourists staying in the community can share experiences in order to obtain more information to be used in possible recommendations to other agents' tourists. In Figure 6 you can see the dialogue that exists between two agents, when a report is released as was explained in the last section.

When there is new enough content on classifications for a given profile of tourists, the Agent Adviser sends news for the community, with the aim of the agents' tourists, who have similar interests, start a communication.

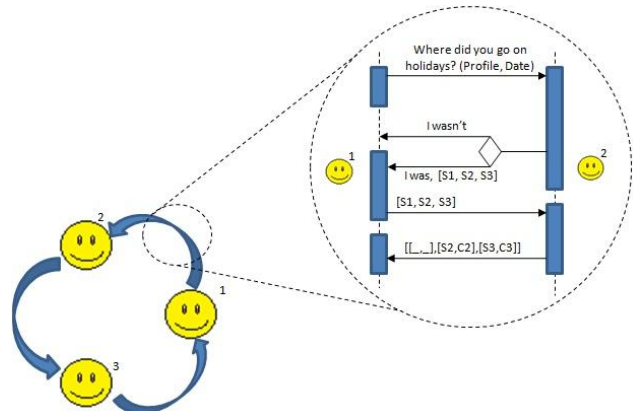


Figure 6 – Agents Tourists Communication Process

Figure 6 represents a dialogue that can exist between Agt1 and Agt2. Agt1 asks Agt2 "Where did you go on holidays?", in a certain date. Agt2 can say that from that date has not yet made any travel or, if it did, send him a message with all the places that it went on vacation after that date. If Agt1 receives a positive response, sends to Agt2 a message with all cities where it was already on

vacation. The Agt2 responds with information about the ranking of cities that Agt1 sent him, if it has already visited. At the end with the information collected, Agt1 will analyze its similarities with Agt2, the similarity level of classification of sites and locations chosen to go on vacation.

To prove how similar are two agents, agent target will measure the similarity between all agents of its kind, individually. So, after knowing which agents can contribute to the process of choosing the destiny of his trip, the target agent calculates the similarity of their ratings with the ratings assigned by other agents. This classification refers to trips that were performed by both. To concretize this goal, we use the concept of Euclidean distance. The Euclidean distance formula allows to calculate the distance between two points where these two points will be constituted by the classifications assigned by the two agents to certain cities visited.

$$d_{euclidean} = \sqrt{\sum_{i=1}^d (a_i - b_i)^2} \quad (2)$$

Table 2. Calculate distance between two points

	New York	London	Paris	Rio Janeiro	Distance
Classification of target tourist	2	1	4	5	-----
Classification of tourist 1	1	3	1	4	3,87
Classification of tourist 2	2	4	1	5	4,24
Classification of tourist 3	4	1	2	3	3,46

In table 2 is shown the distance that an Agt1 has for each of the other agents in respect of classifications awarded to the cities visited by both. The tourist to which the Agt1 has a smaller distance will be one that best reflects a similar type of evaluation.

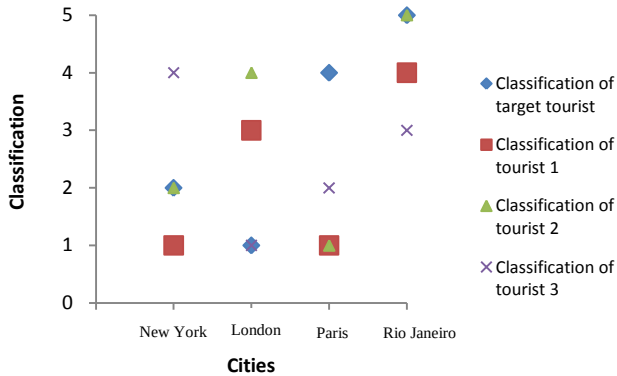


Figure 7 – Graphic of the calculation of distances

In Figure 7 we can get a better visual perception of distance between the rating assigned by the tourist target and the rating of other tourists in each place visited by the tourist target.

After finding out which agents carrying out a classification more similar to the tourist target (thus demonstrating a level of demand and similar assessment), we look for those with an index of demand for similar holiday destinations.

For this calculation, are used two formulas:

- For the total number of trips made by the agent that is used as a comparative element is ≥ 5 :

$$\frac{\text{Travel Identical}}{\text{Travel Performed}} \times \frac{\text{Travel Identical}}{\text{Sum Travel}} = \frac{\text{Travel Identical}^2}{\text{Travel Performed} \times \text{Sum Travel}} \quad (3)$$

- For when the total number of trips made by the tourist that is used as a comparative element is < 5 :

$$\frac{\text{Travel Identical}}{\text{Travel Performed}} \times \frac{\text{Travel Identical}^2}{\text{Sum Travel}} = \frac{\text{Travel Identical}^3}{\text{Travel Performed} \times \text{Sum Travel}} \quad (4)$$

As shown in the used formulas, are considered the total trips made by other tourists, so that we can examine them properly, the similarity in choice of vacation destinations on the chosen. After the calculus the tourist target will increase the level of belief in the tourists that have on their overall travel a greater range of options equal. In table 3 are presented the results of the belief for three tourists.

Table 3. Belief in several tourists

	Travel Identical	Travel Performed	Belief
Tourist 1	5	10	0,0125
Tourist 2	5	5	0,0250
Tourist 3	5	50	0,0025

After knowing which tourists with more similar preferences and then to calculate the confidence of tourists who provide this information, the next step is to combine the two previous measures, to meet the tourist with the most similar options, taking into account the requirements that are most important.

To achieve this objective are used the following formulas:

- $b_i = \frac{x_{ij}}{\max_{ij}} \rightarrow$ maximize the belief; (5)

- $d_i = \frac{\min_{ij}}{x_{ij}} \rightarrow$ minimize the distance; (6)

- $\text{total}(b, d) = kb + kd \rightarrow$ Utility function. (7)

Table 4 shows the results for the example that was used. As you can see in Table 4, although a tourist who has not made a classification very similar to the tourist target, as its index of confidence is quite high, the tourist will be chosen. This is because the formula used in the Utility Function, checking the results of two tests, we conclude that the global is the tourist who target tourists should give more credit.

In the end, the cites that tourist 2 sent the tourist target will be those that will be recommended, should never have been visited, to tourist target.

Table 4. Calculation of the overall assessment

	Belief	Distance	Total
Tourist 1	0,0125	3,87	0,697
Tourist 2	0,0250	4,24	0,908
Tourist 3	0,0025	3,46	0,550

6 Related Work

In order to make a comparison between the system presented and existing systems was realized a brief study about three systems in the tourism area with a high success rate among the community of tourism.

The TripAdvisor[6] is a site that has as main objective the recommendation of travel, locations and activities for each user. This system makes the recommendation to the tourist but based on data entered by the tourist, while the system presented in this paper allows recommending the tourist from the analysis of historical data of the tourist and still examining the history of other tourists who have identical preferences.

Other recommendation system analyzed was the DieToRecs [7]. This system has many similar features to the system discussed earlier, adding to contain a module that allows distinguishing the various types of users: beginners, intermediate and advanced. However, this system also has the problem of the previous system, i.e., recommendation is made only with the aid of the user.

The last recommendation system analyzed was the Expedia [10]. The system enables users to choosing a travel package, ie, choose the flight, hotel, activities to be undertaken, among others. Users can also know the evaluation of the trips made by other tourists.

Nonetheless it would be not correct to classified this one as a recommendation system since it does not return an individual tourist recommendation.

Thus, the system presented in this article allows making a recommendation more complex and crucial to the tourist. As stated above, this system recommends tourists based on their history and also based on historical tourists with similar preferences.

7 CONCLUSION

The system presented by the techniques described may have a huge importance in the process of decision making for tourists in tourism domain.

Assuming that nowadays a tourist is a person with a very busy life and expecting to make the process of choosing a destination to visit spending the shortest time possible, a recommendation system that focuses on these aspects will certainly be a great aid for the tourist. In this proposal the objective is not to set a new system but to create features and techniques that are still underdeveloped in this area.

There is currently a prototype that was developed, able to support and implement the techniques that were described. Our aim is to soon put the prototype online in order to gather information, to validate the concepts presented.

Our main goal is that by using this system, tourists who use it, find and are presented with the best suggestions, the ones that really suit their interests.

We expect that using recommendation system allow to suggest to tourists a ideal recommendations. In some cases recommendations that tourists could not imagine at a first glance.

REFERENCES

1. Page, S., Connell J. – Tourism: a modern synthesis, London, Cengage Learning EMEA, ISBN: 9781844801985, pp. 63-74, 2006.
2. Moutinho, L. - Consumer behavior in tourism, European Journal of Marketing, pp. 21-44, 1987.
3. Sebastia, L., Garcia, I., Onaindia, E., Guzman, C. – e-Tourism: a tourist recommendation and planning application, 20th IEEE International Conference on Tools with Artificial Intelligence, Ohio, USA, 2008.
4. Pejic, A., Pletl, S., Pejic, B. – An Expert System for Tourists Using Google Maps API, 7th International Symposium on Intelligent Systems and Informatics, Subotica, Servia, 2009.
5. Coelho, B., Martins, C., Almeida, A. – Adaptive Tourism Modeling and Socialization System, International Conference on Computational Science and Engineering, Vancouver, Canada, 2009.
6. Santos, R., Marreiros, G., Ramos, C., Neves, J., Bulas-Cruz, J. – Using Personality Types to Support Argumentation, 2009
7. Gibson, H., Yiannakis, A. – Tourist roles: Needs and the lifecourse, Annals of Tourism Research, vol. 29, No. 2, pp. 358-383, 2002.
8. TripAdvisor, 2010, <http://www.tripadvisor.com>
9. DieToRecs, 2010, <http://dietorecs.itc.it>
10. Expedia, 2010, <http://www.expedia.com>

Combining Loan Requests and Investment Offers in Peer-To-Peer Lending

Luís Martinho¹ and Luís Paulo Reis^{1,2}

Abstract. Online Peer-To-Peer lending has seen some growing media attention since its recent creation. Nonetheless, the systems which provide deal brokerage in this context have yet to be given significant consideration within the scientific community. This paper is part of a broader effort to setup a Peer-to-Peer lending community in Portugal. This work focuses on solving the infrastructural problem of combining investment offers from potential lenders with loan requests from potential borrowers. The combination process must strive for an optimal result, which pleases lenders and borrowers alike, despite their opposing agendas. Simultaneously the combination result should also benefit the platform's business model, so as to keep it sustainable and profitable. Several optimization metaheuristics, powered by a constraint programming module, were applied to efficiently explore the problem's solution space and to find optimal solutions. The results achieved with this approach show how metaheuristic-driven optimization can be successfully applied to Peer-to-Peer lending combination problems.

1 INTRODUCTION

Arguably one of the most powerful concepts to emerge from the Internet was that of the social web: a network made not only of machines, but also of people who could now relate directly, no matter how geographically apart. This new interaction paradigm not only challenged existing business models, but also motivated completely new ones. Some of the business models that were most impacted were those that involved intermediation. This was the case with the various forms of employment portals (where employees and employers could meet directly) and auction portals (where buyers and sellers could meet directly). The next step would be direct person-to-person lending or peer-to-peer lending, starting with Zopa in the United Kingdom in February 2005 [12].

This work builds on efforts made from early 2007 to the present day to create an online Social or Peer-To-Peer (P2P) Lending platform, operating in Portugal. Peer-To-Peer lending being understood as lending and borrowing, directly between individuals ("peers") without the participation of a traditional financial institution.

The success of the project was seen as greatly dependent on the individual satisfaction of both lenders and borrowers, despite their opposing agendas. It was thus required to create a mechanism that could combine loan requests and investment offers in a fashion which pleased the greatest amount of participants, while protecting the interest of the platform operator. It was this key problem that motivated

this paper.

The main objective of this work was then to build a system capable of successfully finding optimal combinations of loan requests – defined by the amount requested and the maximum rate at which the potential borrower is willing to repay the money – and investment offers – defined by the amount offered by the lender and the minimum interest rate at which the potential lender is willing to receive its money back. Simultaneously, the system should attempt to maximize the amount of money traded, due to the volume based business model of the project.

The rest of this paper is organized as follows. The next section formulates the combination problem extensively, to explain the relevant inputs for the combination system. Section 3 provides a short overview of well known optimization metaheuristics, that were found relevant for the work at hand. Section 4 details the system design, used to create the metaheuristic optimization framework and to apply it to the loan matching problem. Section 5 presents the results achieved using the developed system. Finally section 6 discusses the obtained results and highlights future research paths for this work.

2 COMBINATION PROBLEM FORMULATION

The problem of combining loan requests and investment offers, can be considered as an optimization problem, taking the decision variables as the rates at which loans are matched, together with the amounts involved. The constraints would be set by the conditions specified by the members, when placing their terms for intended rates and amounts. The objective function would take into account the stated goals of the platform: to maximize the satisfaction of both borrowers and lenders, while contributing for the platform's profitability.

2.1 Formal Problem Definition

More formally the problem $P = (S, f)$ can be defined as a generic optimization problem by specifying the set of parameters:

- N is the number of lenders participating;
- M is the number of borrowers participating;
- R_{min_i} is the minimum rate at which lender i is willing to lend its money;
- R_{max_j} is the maximum rate at which borrower j is willing to borrow money;
- A_{max_i}, A_{min_i} are the maximum and minimum amounts of money, lender i is willing to lend;
- A_{max_j}, A_{min_j} are the maximum and minimum amounts of money, borrower j wants to borrow.

¹ Informatics Engineering Department of the Faculty of Engineering of the University of Porto, Portugal, email: luis.martinho@fe.up.pt

² Laboratory of Artificial Intelligence and Computer Science of the University of Porto, Portugal, email: lpreis@fe.up.pt

The set of the decision variables $X = r_{11}, a_{11}, r_{12}, a_{12}, \dots, r_{ij}, a_{ij}, \dots, r_{NM}, a_{NM}$, where:

- r_{ij} represents the rate at which lender i lends to borrower j and
- a_{ij} represents the amount of money lender i lends to borrower j .

The variable domains defined for each instantiation of the matcher where:

- $D(r_{ij})$ is the interval $[0, 1]$, where the maximum allowed rate is 100%, due to business considerations;
- $D(a_{ij})$ is the interval $[0, A_j]$, where A_j represents the maximum amount requested by borrower j .

The constraints which result from the members' terms, which consist of:

- $C1 : \frac{\sum_{i=1}^N a_{ij} r_{ij}}{\sum_{i=1}^N a_{ij}} < R_{max_j} \forall j \in 1, \dots, M$, i.e., the overall loan rate for borrower j must be less than or equal to the proposed maximum rate;
- $C2 : \frac{\sum_{j=1}^M a_{ij} r_{ij}}{\sum_{j=1}^M a_{ij}} \geq R_{min_i} \forall i \in 1, \dots, N$, i.e., the overall investment rate for lender i must be greater than or equal to the proposed minimum rate;
- $C3 : A_{min_i} \leq \sum_{j=1}^M a_{ij} \leq A_{max_i} \forall i \in 1, \dots, N$, i.e., the total amount invested by the lender i must be inside the amount range offered by the lender;
- $C4 : A_{min_j} \leq \sum_{i=1}^N a_{ij} \leq A_{max_j} \forall j \in 1, \dots, M$, i.e., the total amount received by borrower j must be inside the amount range requested by the borrower.

A function $f : D(r_{11}) \times D(a_{11}) \times \dots \times D(r_{ij}) \times D(a_{ij}) \times \dots \times D(r_{NM}) \times D(a_{NM}) \rightarrow \mathbb{R}$, which computes the utility to maximize, defined as mapping the decision variable's domains into a decimal value.

The solution space is then the set of valid solutions, or: $S = \{s = \{\dots, (r_{ij}, R_{ij}), (a_{ij}, R_{ij}), \dots\} : R_{ij} \in D(r_{ij}) \wedge A_{ij} \in D(a_{ij})\}$, where s satisfies all constraints.

2.2 Utility function

Defining an utility function is a complex issue and has a great impact on the overall business quality of the solutions. Nonetheless validation of such a function presents a non-trivial challenge, which may fit better the field of efficient markets or an even more social science domain, rather than the field of computer science. This is why the comparative study of different utility functions was deemed out of scope for this work.

The "tight margin utility" (TMU) function was created to reward profitable and fair solutions. Its base rationale is that people specify their limits as their baseline, from which they expect some sort of profit margin. To achieve this, the function takes into account three key factors:

- member profit: how much did the members benefit from the solution, what was their margin;
- fairness: how evenly distributed are these benefits;
- demands satisfaction: how well where the demands fulfilled in the solution, this is particularly significant for the platform's profitability which comes from the volume of money successfully matched.

More formally the TMU is computed using equation 1.

$$TMU = (k_{MM} \times MM) \times (k_T \times T) \times (k_{FR} \times FR) \quad (1)$$

Where:

- MM (Member Margin) represents the sum of margins obtained by each member. Here margin is understood as the difference between the initial terms the member specified and the actual deal conditions proposed by the solution. This can be computed using equation 2.

$$MM = \sum_{i=1}^N (r_{ij} - R_{min_i}) + \sum_{j=1}^M (R_{max_j} - r_{ij}) \quad (2)$$

- T (Tightness) represents how close together are the individual gains of each member, a solution where some member make are highly favored in the detriment of others exhibits low tightness where a solution where margins are homogeneous is considered tight. Tightness can then be computed using 5, which uses the standard deviation of the set of member margins, taking into account both the lender margins (LM) and borrower margins (BM) sets, as computed in 3 and 4, respectively.

$$LM = \{(r_{ij} - R_{min_i}) : i \in [1, N] \wedge j \in [1, M]\} \quad (3)$$

$$BM = \{(R_{max_j} - r_{ij}) : i \in [1, N] \wedge j \in [1, M]\} \quad (4)$$

$$T = \frac{1}{\sigma(LM \cup BM)} \quad (5)$$

- FR (Fulfillment Rate) indicates the relevance of the amount that was matched, in the light of the total amounts of money which were in stake. A fulfillment rate of 1 or 100% means that all the available money was invested and all the money requested was fulfilled. This can be computed using equation 6.

$$FR = \frac{2 \times TotalMatchedAmount}{(TotalOfferedAmount + TotalRequestedAmount)} \quad (6)$$

- *TotalMatchedAmount* is the total amount that was successfully matched between lenders and borrowers; *TotalOfferedAmount* is the total amount that was offered by lenders; *TotalRequestedAmount* is the total amount that was requested by borrowers.
- Each of the factors is weighed according to a specific parameter (k_{MM} , k_T and k_{FR}), which allows to fine tune the utility function for the kind of results intended.

3 OPTIMIZATION METAHEURISTICS

The term metaheuristic, first introduced by Glover [4] describes solution methods that mix higher level strategies with local improvement procedures in order to escape from local optima and to perform a robust search of a solution space. Today it refers to a broad class of strategies for optimization and problem solving.

The techniques described here are all based upon the idea of choosing a starting point and then altering one or more variables in an attempt to increase the fitness or reduce the cost. The various approaches can be split across the following in two groups: trajectory and population-based metaheuristics [6].

Trajectory-based metaheuristics, which usually use a single candidate solution, comprise both methods which only exploit locally and those which combine exploration and exploitation:

- **Pure Random Search:** the simplest global random search algorithm. It consists of taking a sample of n independent random points and evaluating the fitness function for each of them. Not only very simple to implement, it is often used as a benchmark for comparing properties of other global optimization algorithms [15].
- **Hill-Climbing:** the name hill-climbing implies that optimization is viewed as the search for a maximum in a fitness landscape. However, the method can equally be applied to a cost landscape, in which case a better name might be valley descent. The algorithm is easy to implement, but is inefficient and offers no protection against finding a local minimum rather than the global one. From a randomly selected start point in the search space, i.e., a trial solution, a step is taken in a random direction. If the fitness of the new point is greater than the previous position, it is accepted as the new trial solution. Otherwise the trial solution is unchanged. The process is repeated until the algorithm no longer accepts any steps from the trial solution.
- **Simulated Annealing:** originally described by Kirkpatrick in [10], Simulated Annealing (SA) tries to emulate the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and compares this process to the search for a minimum in a more general system. The SA algorithm tries to escape local optima by allowing the search to sometimes accept worst solutions with a probability (p), which decreases along with the temperature of the system (t).
- **Tabu Search:** tabu search is based on the premise that problem solving, in order to qualify as intelligent, must incorporate adaptive memory and responsive exploration. An analogy provided by [5] is that of mountain climbing, where the climber must selectively remember key elements of the path traveled (using adaptive memory) and must be able to make strategic choices along the way (using responsive exploration).

Population-based metaheuristics, are those which maintain a set of candidates, and usually explore the solution space freely, in order to escape local optima:

- **Evolutionary Computation (EC):** the family of algorithms which share the metaphor of natural evolution, loosely adapted from the field of biology. The underlying concept is that, given an initial population, by selecting only the fittest elements to survive and reproduce, one should expect that each new generation of offspring generates fitter individuals. With this principle in mind a series of methods have been developed, most notably: Genetic Algorithms (GA), Evolution Strategies (ES) and Evolutionary Programming (EP). For the interested reader, a more detailed taxonomy of EC algorithms is available in [2].
- **Particle Swarm Optimization (PSO):** according to its authors in [9], particle swarm optimization (PSO) has its roots in two main component methodologies. Perhaps more obvious are ties to artificial life in general and bird flocking, fish schooling and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolutionary programming. The basic underlying concept is the way how social learning influences our beliefs and behaviors. For a given problem, with a known fitness function, a population of individuals, or particles, defined as random solutions for the problem is initialized. An iterative process to improve these candidate solutions is set in motion. The particles iteratively evaluate the fitness of the candidate solutions and remember the location where they had their best success. The individual's best solution is called the par-

ticle best or the local best. Each particle makes this information available to their neighbors. They are also able to see where their neighbors have had success. Movements through the search space are guided by these successes, with the population usually converging, by the end of a trial, on a problem solution better than that of non-swarm approach using the same methods [8].

3.1 Constraint Programming Libraries

The constraint programming paradigm can be used together with a number of other paradigms. Nonetheless, the need to easily integrate the matching system produced by this work, with a separate Web application suggested the need for a language that was dynamic, to allow the necessary interactive experimenting, had a strong base library to facilitate integration with the Web system so that the overall development time was shortened. The choice was narrowed down to Ruby, since the existing Web application had been built using Ruby on Rails [14] and Python due to its very mature base library and extensive multi-paradigm support (which would make for smoother connection between the object-oriented and the constraint programming paradigms used).

The choice was `python-constraint` essentially due to its pure Python nature, its small size (circa 1400 library lines of code, including comments and whitespace) and the quality of the documentation available in [11]. These characteristics improve the learning experience and open the way to future extension.

4 METAHEURISTICS BASED OPTIMIZATION FRAMEWORK

This section details the system design, used to create the metaheuristic optimization framework and to apply it to the problem at hand. The system consists of three components: a constraint solving tool; the optimization suite and an adapter for the suite that uses the constraint solving library to solve the specific problem. Figure 1 illustrates these blocks and shows their high-level interactions.

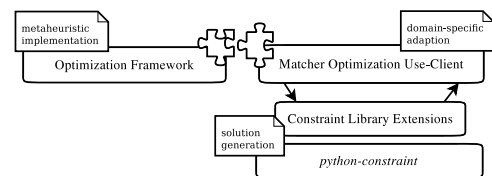


Figure 1. Overview of the solution architecture

4.1 Constraint library extensions

The first problem to address was that of efficiently producing feasible solutions, complying with all the existing constraints. The extensions made to the base `python-constraint` library intend both to provide additional constraints and to add functionality required by a large number of meta-heuristics which would depend on the solution generation mechanism to explore the solution space. The added constraints were necessary for the correct specification of the optimization problem, as formulated earlier. This was the case with providing constraints on weighted averages, as required by the member rate terms, both for lenders and borrowers, which are weighed by the amount involved in each contract. From the optimization standpoint,

there were low-level functionalities that required close interaction with the solution generation strategy, yet were relevant for solution space exploration. This was the case with an API method for retrieving the valid neighborhood for a given solution, as well as a way for obtaining the closest feasible solution to a specified solution, even if invalid. These operations could not be efficiently implemented at a higher-level or would otherwise require replicating a large part of the solution generator behavior in the optimization adapter layer.

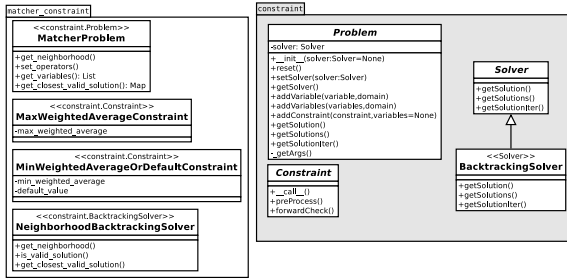


Figure 2. Constraint library and extensions architecture

Figure 2 illustrates how the constraint library extensions module extends the `python-constraint` library, depicted in grey background. This module provides additional functionality which is relevant for the current problem domain. The features include neighborhood generation, useful for several search meta-heuristics, getting approximate valid solutions from an initial solution, whether or not valid, and new constraints to enforce the members' terms as formulated in section 2.1. Below is a description of the main object classes introduced, and the purpose they serve:

- **Problem wrapper:** The `MatcherProblem` class extends `constraint.Problem`, the `python-constraint` façade class used to define a problem and retrieve solutions from a solver [11]. Introduces the `getNeighborhood` method which accepts a solution, in the form used throughout the constraint library, and returns a list of the adjacent solutions produced. The problem wrapper also provides the `getClosestValidSolution` method which returns a valid solution, closest to a specified solution, whether or not valid. This feature is relevant for search meta-heuristic that potentially generate invalid solutions which must be converted to valid ones, this is the case with evolutionary approaches such as genetic algorithms.
- **Solver:** a new solver was developed that provided new solution space exploration capabilities. The `NeighborhoodBacktrackingSolver` extends the `constraint.BacktrackingSolver` provided by `python-constraint`, which provides a constraint solver with backtracking capabilities.
- **Constraints:** In the addition to maximum and minimum sum constraints, two new constraints were developed for the constraint library: maximum weighted average constraint and minimum weighted average constraints. Both of these constraints are relevant for restricting the average rate a member gets in the final match result, since the rate is weighed by the amount of money in each match.

4.2 Optimization framework

The optimization framework provides strategies for solving optimization problems. The optimization framework specifies the abstract behavior of an optimizer object, and provides concrete implementations of popular search meta-heuristics such as those described in section 3. The problem specific use-client code needs only to take care of what is really domain specific [13]: the solution generation, solution evaluation and solution visualization.

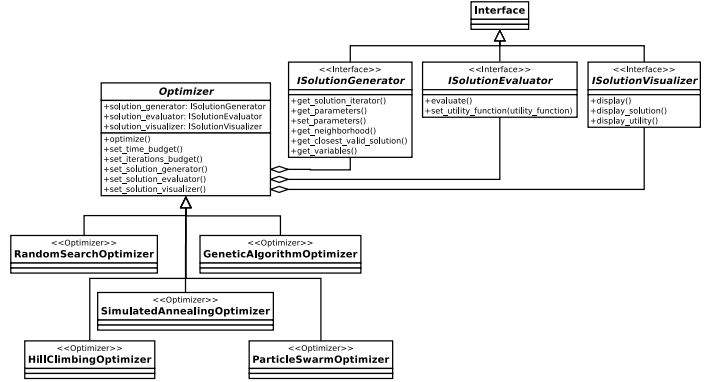


Figure 3. Optimization framework architecture

The `AbstractOptimizer` class holds the search strategy used to optimize the problem, using the provided generator and evaluator. Subclasses of this abstract optimizer template provide the concrete implementations of each meta-heuristic. The abstract class already defines basic termination management code which can be reused or augmented by child classes.

4.3 Matcher optimization use-client

The problem-specific adapter bridges the two components described above. An optimization adapter is placed inside the inversion of control framework, providing solution generation, evaluation and visualization services to the `Optimizer` classes. The adapter provides the solution generation infrastructure introduced above, as well as solution evaluation services, implementing the utility function.

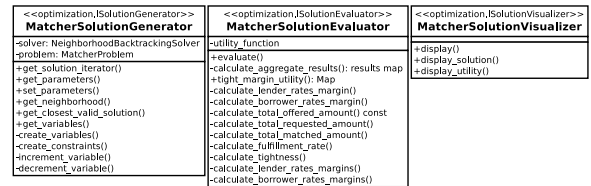


Figure 4. Matcher optimization use-client architecture

The two main components of the use-client are:

- **Solution Generator:** the `MatcherSolutionGenerator` class generates matcher solutions for the configured problem, according to a defined parameter specification for credit matching problems. It uses the matcher constraint library as the solution generation mechanism.

- **Solution Evaluator:** the solution evaluator built for the matcher adapter is `MatcherSolutionEvaluator`. It evaluates solutions according to a selected utility function. It aggregates the standard format solution into a final result map, similar to the problem parameters. It then evaluates the configured utility function, specifying the problem's parameters and the resulting matches for the specified solution.

5 EXPERIMENTAL RESULTS

A series of experiments were designed to test and validate the solution described in the previous sections. This section describes how the experiments were assembled, and how the solution behaved in different scenarios and using different internal strategies.

5.1 Experimental Scenarios

A key concern while testing the solution was to have a sufficient amount of quality data that could be used to exercise the solution under different conditions, but maintaining a set of constant parameters. The solution was to design a simple data generator which would receive a high-level specification of the test scenario, and would then, stochastically, create a complete dataset to use as input for the matcher system.

To cater to the different types of settings, the following parameters stood out as relevant:

- N and M , the number of lenders and borrowers, respectively;
- $\bar{r}_i$, the mean lender rate;
- σ_{r_i} , the lender rate standard deviation;
- $\bar{a}_i$, the mean lender amount;
- σ_{a_i} , the lender amount standard deviation;
- $\bar{r}_j$, the mean borrower rate;
- σ_{r_j} , the mean borrower standard deviation;
- $\bar{a}_j$, the mean borrower amount;
- σ_{a_j} , the borrower amount standard deviation.

This information comprehensively describes a scenario, i.e., a template containing a high-level definition of the environment parameters. Using this type of templates, with the help of a stochastic environment generator – which was developed on top of a statistics module for Python [1] – it is possible to generate different matching environments each time a run is made.

The two scenarios which define the input for the different experiments were as follows:

1. **Tight market:** lenders offer lower interest rates, with borrowers allowing higher rates when finding a loan. Small deviation for both lenders and borrowers, keeping the market homogeneous.
2. **Loose market:** offer lower interest rates, with borrowers allowing higher rates when finding a loan. Larger deviation in rates distribution for both lenders and borrowers, providing some diversification and heterogeneity.

Table 1 shows the exact settings for each of the referenced scenarios.

The scenarios were used as background for the experiments conducted, and allowed to understand how the different strategies and settings behaved under different environment conditions.

Table 1. Detailed scenario settings.

Parameter	Scenarios	
	Tight market	Loose market
N	5	5
M	5	5
$\bar{r}_i$	5.00%	5.00%
σ_{r_i}	1.00%	5.00%
$\bar{a}_i$	1,000.00	1,000.00
σ_{a_i}	100.00	100.00
$\bar{r}_j$	20.00%	20.00%
σ_{r_j}	20.00%	20.00%
σ_{r_j}	1.00%	5.00%
$\bar{a}_j$	1,000.00	1,000.00
σ_{a_j}	100.00	100.00

5.2 Metaheuristic analysis

The performance of each metaheuristic on the utility function was measured by successive executions, under different environments as mentioned above. Using the prepared scenario templates, an environment was generated for each type. Under each scenario, the metaheuristics were executed with a fixed 1000 iteration budget. The result of the utility function in each run was then recorded, and the experiment repeated using the same settings to record the mean utility score. In the following section, the results of the successive executions are presented and analyzed.

- **Scenario 1 – Tight market:** the graph in Figure 5 presents the results for the utility function under optimization at each iteration during the algorithm runs. The utility values are plotted for each of the implemented metaheuristics. The results suggest that the most efficient method was GA, converging towards better results considerably faster than the other algorithms and finishing with a better overall result after the all the runs were completed. The evolution is nonetheless irregular, exhibiting relevant breakthroughs at certain iterations, suggesting the importance of certain exceptional mutations and crossovers, as opposed to a continuous evolutionary improvement. With a more regular progression is the PSO algorithm, with strong results appearing late (by iteration 400) but keeping a consistent progression until stagnation at a local optima. The HC algorithm displays an interesting, near linear, improvement of the utility score as iterations increase. The PRS approach had very bad results, with the solution space exploration mechanisms trapping the metaheuristic in a region with bad solutions. The PRS did not offer a relevant contribution to the specific scenario, as it did not provide a reasonable baseline. The results indicate that the baseline could be considered to be the SA approach due to its below average performance in this environment.
- **Scenario 2 – Loose market:** The graph in Figure 6 presents the results for the utility function under optimization at each iteration during the algorithm runs. The utility values are plotted for each of the implemented metaheuristics. Notably the best performance also belongs to GA, although, in this setup, the HC and even the PRS have interesting performances. Near the end of the iteration budget, these two trajectory based algorithms surpass the results of the PSO implementation.

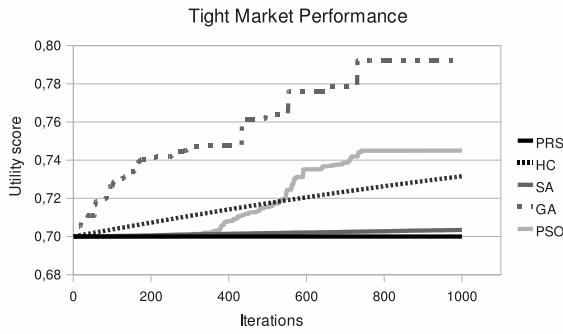


Figure 5. Scenario 1 – Plot of utility function value against iterations (by metaheuristic)

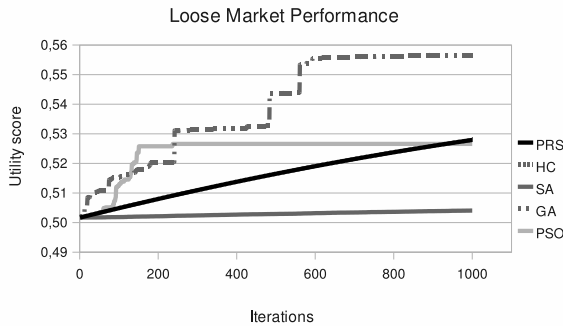


Figure 6. Scenario 2 – Plot of utility function value against iterations (by metaheuristic)

6 DISCUSSION AND CONCLUSIONS

From the experiments presented in section 5, it results that the system is effectively performing successful matches, and it is possible to pick the best performing metaheuristics in each scenario to use in a production environment. To that extent the goal of building a system capable of successfully finding optimal combinations of loans requests and investment offers, is believed to have been reached. The optimization approach worked and was successful in exploring the solution space considering the provided utility function. The optimization framework is also seen as of value in itself. The framework was flexible enough to support multiple metaheuristics, gradually added as the work progressed, and was adequately decoupled from the problem to be used in solving other test problems. This level of modularity also allowed the problem specific code to be evolved by itself, while keeping the API contract, without breaking the optimizer as a whole.

6.1 Future Work

- **Utility Alternatives:** Having seen the various optimization strategies applied to the utility function, it seems that the solution is effectively verified, i.e., solving the problem right. Nonetheless the question remains if the current solution solves the right business problem, i.e., a question of validation. Relevant and important as it would be, this study would still be closer to the subject of Efficient Markets in the field of Economics, and was considered out of scope for this work.
- **Distributed Framework:** despite carefully designed and effectively reusable, the optimization framework left out of its scope a sig-

nificant topic in optimization, which is distribution. An interesting direction for this work, could be leveraging existing distribution strategies such as MapReduce (see [3] for an introduction to MapReduce and [7] for a PSO implementation based on the MapReduce programming model).

- **Multi-Agent System Testbed and Matcher:** another approach which could bring significant insight into the problem, would be that of multi-agent systems. For one, building a multi-agent system that simulated the problem, with agents modeling lenders and borrowers with distinct profiles, would provide an interesting testbed for the existing matching system. Additionally, if negotiation mechanisms were to be added to the multi-agent platform, one could expect to achieve an alternative matching system which would mimic more closely the underlying phenomena of Peer-to-Peer lending.
- **Social Rating:** the final point to explore, would be that of trust management in the Peer-to-Peer lending community. Existing P2P lending platforms rely on external trust anchors to determine the reliability of members. This is, by nature, a centralized approach which does not honor the social emphasis of a P2P network, and implies additional costs for both the platform operators and the users. The trust management angle to this solution would imply designing a social rating scheme, based on work done in the fields of collaborative filtering and trust management, which allowed users to get involved in evaluating other users they knew and trusted.

REFERENCES

- [1] I. Albert and G. Strangman, *python-statlib – Google Code*, <http://code.google.com/p/python-statlib/>, retrieved in June 2009, (2007).
- [2] P. Cálegary, G. Coray, A. Hertz, D. Kobler and Pierre Kuonen, ‘A taxonomy of evolutionary algorithms in combinatorial optimization’, *Journal of Heuristics*, 145–158, (1999).
- [3] J. Dean and S. Ghemawat, ‘MapReduce: Simplified Data Processing on Large Clusters’, *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, 107–113, (2004).
- [4] F. Glover, ‘Future paths for integer programming and links to artificial intelligence’, *Computers and Operations Research*, 533–549, (1986).
- [5] F. Glover and M. Laguna, *Tabu Search*, Springer, 1st edn., (1997).
- [6] A. Hopgood, *Intelligent Systems for Engineers and Scientists*, CRC Press, 2nd edn., (2001).
- [7] A. W. McNabb, C. K. Monson and K. D. Seppi, ‘MRPSO: MapReduce particle swarm optimization’, *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 177–177, ACM New York, (2007).
- [8] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, (2001).
- [9] J. Kennedy and R. C. Eberhart, ‘Particle Swarm Optimization’. *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, 19421948, IEEE, (1995).
- [10] S. Kirkpatrick, C. Gelatt and M. Vecchi, ‘Optimization by Simulated Annealing’, *Science*, 671–680. New Series, Vol. 220, no. 4598, (1983).
- [11] G. Niemeyer, *API Documentation (python-constraint)*, <http://labix.org/doc/constraint/> retrieved in May 2009, (2005).
- [12] The Register, ‘Zopa – the bank that likes to say ‘Hello There!’’, *The Register*, (2005).
- [13] D. Riehle, *Framework Design: A Role Modeling Approach*, Ph.D. Thesis, no. 13509, ETH Zürich, (2000).
- [14] D. Thomas, D. H. Hansson, *Agile Web Development with Rails*, The Pragmatic Programmers, 3rd edn., (2009).
- [15] A. Zhigljavsky, *Stochastic Global Optimization*, Springer, (2007).

In the Search of Better Deals Using Trust

Joana Urbano and Ana Paula Rocha and Eugénio Oliveira¹

Abstract. In current days, a great effort, on several scientific research areas, is being devoted to the automation of the procurement processes in the framework of business-to-business relations. However, several constraints limit the extension of the procurement process to truly open and global marketplaces. One such constraint is the lack of valid trust mechanisms that allow business agents to select partners outside the sphere of known relationships, in the hope of better deals. Following our firm belief that trust systems used in such hard environments shall be situation-aware and flexible, we present our approach for such a system. Also, we set up a multi-agent environment based on client-supplier transactions in order to evaluate two fundamental questions. First, whether the break of parochial relations allows for better deals without jeopardizing the overall utility of client agents by engaging on transactions with strangers. And second, how representative trust systems – including our own approach – support the exploration of new partners in a relative safe way.

1 INTRODUCTION

In current days, a strong I&D effort in different research areas is being put in the automation of the procurement processes of business-to-business relations. Ideally, the development of adequate agreement technologies would allow opening the space of business opportunities, where potential partners have the means and the confidence knowledge to search for good business opportunities outside their limited sphere of breeding relations.

However, there are actually constraints that limit this desired openness, and the fear of risking unknown partners is one of the biggest barriers to trade in a truly open market environment. In fact, economic exchanges with strangers can result in harm for the intervenient agents in both ways. Concerning a subcontracting scenario in the textile industry as an example, the client part of the relation can be deceived by the provider part in several different ways [1]:

- A delay in delivery, which affects all the supply chain;
- The quality received, as specified by affordability, safety and degree of uniqueness parameters;
- The quantity received (too much or too less);
- The violation of intellectual property rights;
- Ethical problems;
- Other problems, such as price rise and legislative changes.

Without some kind of a trust mechanism, it is reasonable to conclude that business partners would preferentially adopt parochial environments in detriment to more aggressive exploration of deals outside the already known partner relationships space. For instance, in the fashion retail industry, clients often rely on knowledge available through textile fairs and textile agents to make the bridge between brands and the trustable and reliable textile suppliers. However, even with this form of trust guarantees, the space of available suppliers is relatively small and strongly supported on the expected behavior of the partner, rather than on the real utility of the business transaction.

This paper addresses the need to develop computational trust systems that can be used in open and global markets, where the evidences that can be used to infer the trustworthiness of business agents are scarce, contextual and heterogeneous. It is our firm belief that these systems, in order to be acceptable, shall present the following characteristics:

- The aggregation of trust evidences into trustworthiness scores shall follow important properties of the dynamics of trust, as addressed in research areas related to social sciences and psychology;
- The trust inference process shall be fed from diversified and heterogeneous sources of information, such as past direct experiments, available reputation, specific recommendations, perceived roles, and credit ranking agencies;
- The trust system shall be able to detect different situations, and also to detect tendencies of agents' behavior in the presence of such different situations;
- The trust system shall be able to infer the trustworthiness of agents even where the number of trust evidences is scarce.

In this paper, we set up a multi-agent simulation environment where textile client agents seek for optimal deals by selecting from a range of suppliers with different behaviors. Particularly, we address two main questions. First, we evaluate whether the break of the breeding parochial relations between business partners permits to increase the overall utility of clients or, on the contrary, jeopardize it as a consequence of the risk introduced by this strategy.

Then, we present our own approach to a trust system that was designed taking into consideration the hard requirements of open markets. Moreover, we evaluate how different trust methods, including our proposal, can support the exploration of new potential partners in such a way that the risk associated to trading with strangers is decreased by the method.

A third achievement of the work presented in this paper is the development of a simulation environment that can be used to support important studies about parochialism and trust that are being done in the social sciences area, such as the ones presented in [2] and in [3].

The rest of this paper is structured as follows. Section 2 presents our trust method, with particular emphasis on the Contextual Fitness component. Section 3 is devoted to the experiments we performed in order to evaluate the issues referred above. Finally, section 4 concludes the paper.

2 OUR CONTEXT-BASED TRUST METHOD

We developed a trust method envisioning its use in future automatic and open business-to-business markets, taking into special consideration the exigent requirements of such environments. Namely, we are concerned with the performance of the method when the trust evidences available about a given target agent are scarce and heterogeneous, and when the activity of the agents under evaluation can span through different situations and contexts.

The current implementation of our system that encompasses the proposed method is composed of two different modules, as depicted in Figure 1:

¹ LIACC – Laboratory for Artificial Intelligence and Computer Science, Faculdade de Engenharia da Universidade do Porto – DEI, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal, email: joana.urbano@fe.up.pt, arocha@fe.up.pt, eco@fe.up.pt

- The *aggregator* component, which is responsible for aggregating the available trust evidences of an agent into a trustworthiness score for this agent. Several trust engines that are defined in the literature can be used (some of the more representative models are reviewed in [4]), although we are particularly interested on engines that model the dynamics of trust, as those described in [5] and [6], as they appear to perform better than the traditional statistical approaches;
- The *Contextual Fitness* component, which tunes the outcome of the aggregating step by taking into consideration the *specificities* of the current business opportunity and the *adequacy* of the target agent to the specific situation under assessment.

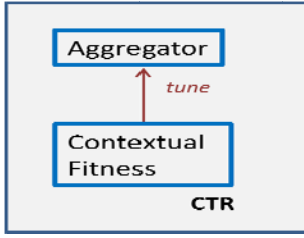


Figure 1. The current implementation of our trust system

The idea beyond this extension is that if the trust system detects that a target agent has some kind of *handicap* related to the current necessity, the overall trustworthiness of the agent for this necessity will be zero; otherwise, the trustworthiness score is the value computed by the simple trust aggregator for the same situation. One good characteristic of this approach is that it can be used with any conventional trust aggregation engine, being it based on statistical, probabilistic or heuristic models, as it is the case of those reviewed in [4].

Before we describe the Contextual Fitness component in more detail, we first introduce the notation and the scenario used all over this paper.

2.1 Scenario and Notation

In the context of this paper, we define $trust_{Ac}(As) \in [0, 1]$ as the trustworthiness value of a trustee agent As , in the eye of a trustier agent Ac , as computed by a traditional trust aggregator engine; and $adequacy\ trust\ ad(As, at) \in \{0, 1\}$ as a binary operator for situation-awareness purposes, where:

- $Ac \in C$ is an agent from the set C of client agents;
- $As \in S$ is an agent from the set S of supplier agents;
- $at \in AT$ describes the *need*, i.e. an instance of the space AT of all possible combinations of attribute-value pairs that describe the need (good, product or service).

In the textile scenario mentioned in the introductory section, a need is announced through a call for proposals issued by a client, and concerns the delivery of some quantity of a fabric due in some delivery time. Thus, an example of an instance of the AT space is the following: {fabric='cotton', quantity='900000', delivery time='15'}. It is worth to note that, for the sake of scalability, all quantitative values are previously quantified using fuzzy techniques.

Therefore, the trustworthiness value of agent As as seen by agent Ac in the specific context at is given by the following equation:

$$trust_{Ac}(As, at) = trust_{Ac}(As) * ad_{Ac}(As, at) \quad (1)$$

This is the same as to say that, in a given moment, an agent may

be qualified as trustworthy in some situation and as untrustworthy in a (maybe slightly) different situation.

Finally, a *contractual evidence* represents a transaction that took place between a client agent Ac and the selected partner agent As , for which an outcome $o \in \{true, false\}$ is generated. I.e., agent As either succeed to provide the good in the contractual terms or violate the contract. Each supplier agent As will, therefore, have an history of its past contractual evidences, each one represented by the tuple $\langle Ac, As, at, t, o \rangle$, where t is the timestamp of the transaction, needed when in use with aggregation systems that weight evidences by their recency.

2.2 The Contextual Fitness Component

The Contextual Fitness component is based on an online, incremental and flexible technique of behavior tendencies extraction that we have developed. Although we have been testing different methods for extracting these tendencies from the historical set of agents' evidences, our current version uses the *information gain* metric. This is a metric used in the machine learning area (such as in the simple decision tree learning algorithm ID3 [7]) for classification purposes. It is typically used as an offline process, implying that the training and testing phases occur before the actual classification of new instances is performed.

The information gain metric is based on the *entropy* concept of information theory, and is defined as following:

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|s_v|}{|S|} Entropy(s_v) \quad (2)$$

where $Gain(S, A)$ is the information gain of attribute A relative to a collection of samples S , $Values(A)$ is the set of all possible values for attribute A , and s_v is the subset of S for which attribute A has value v ([7]).

In our approach, we use this metric to dynamically learn a decision tree from the history of evidences of a given agent As , every time it is necessary to verify the adequacy of the agent proposal to the current need announced by the client agent. In fact, we use all the evidences available about the supplier to build the decision tree, which normally consists of a dataset with some dozens of evidences, if that much. This means that no training or testing phases are performed. After that, the failure tendencies of the agent in evaluation are extracted from the rules pointing to false outcomes. Figure 2 depicts a decision tree that was learned for a given supplier in a specific experiment we have run (we use the Weka API [8] in our simulations).

Concerning the tree below, our algorithm was able to identify that, at the time of this particular assessment, the supplier showed a tendency to fail contracts that match the tendencies ($good = cotton, *, dtime = low$) and ($good = voile, *, *$). Therefore, the trustworthiness value $trust_{Ac}(As, at)$ of agent As , as given by Equation 1, would be zero if situation at matched any of the tendencies derived from the learned decision tree. Otherwise, the trustworthiness value of the target agent for the considered situation would be given by the $trust_{Ac}(As)$ component of equation 1.

Several issues may arise from the use of the information gain criteria in our technique, such as the need to prune the generated trees or the need to use similar metrics that permit heterogeneous evidences (e.g. the gain ratio metric presented in [9]). We address the first question in [10], and leave the second one to future work. On the other hand, we are interested, in this paper, in evaluating the adequacy of our technique when applied to open market environments, where clients risk trading with suppliers that reside outside the space of the clients' breeding environment.

```

good = cotton
|   dtype = low: false
|   dtype = medium: true
|   dtype = big: null
good = chiffon: null
good = voile: false

```

Figure 2: Decision tree generated in our simulations

3 EXPERIMENTS

We set up a multi-agent simulation scenario where business clients in the textile industry try to select the best suppliers of textile fabric, i.e. the ones that would maximize the utility of the clients. In the experiments, we generate a population of clients that have different perspectives concerning the selection of suppliers, depending if it is within the space of their embedded relationships or outside it. Moreover, all suppliers have different handicaps on performing some particular aspect of a business transaction. For instance, some suppliers tend to fail to deliver fabric in short delivery dates, while others might fail to deliver high quantities of any fabric type. The aim of these experiments is to evaluate whether clients that better explore the space of available suppliers would achieve, in the end, higher utility than the clients that adopt a parochial, conservative, strategy, and how a particular trust aggregation technique can better assist this decision.

In this paper, we run two different sets of experiments, as described in the following sections. In the next section, we describe the generic configuration common to both sets of experiments.

3.1 Generic Configuration

Table 1 presents the configuration parameters that are common to both sets of experiments.

Table 1. Configuration of the experiments

Parameter	Value
Fabrics	{Chiffon, Cotton, Voile}
Quantities	{Low, Medium, High}
Delivery Time	{Low, Medium, Big}
# buyers	20
# of sellers	50
Seller stocks	4 contracts per round
Types of sellers	Uniform distribution over the types considered in population
# rounds	60
# runs/ experiment	15

In all experiments, every client selects a supplier that is expected to maximize the utility of the client, where the notion of utility is specific of each set of experiments, as described in the next two sections. It is assumed that every supplier is able to provide any different type of fabric. Also, every client is initialized with a specific business need, represented in a form of a call for proposals (cfp) that is an instance $at \in AT$, as described in section 2.1. The clients keep their need constant in all the experiment rounds. Moreover, all values of possible quantities and delivery times are fuzzified in the categorical values depicted in Table 1.

As for the suppliers, as in order to simulate realistic behaviors, they are initialized with a specific behavior that denotes a specific handicap in specified contexts. For example, a supplier initialized with the *HQT* behavior (standing for ‘Handicap in Quantity’) has a handicap in providing high quantities of any fabric; this way, if it is selected to a business transaction that involves the delivery of high

quantities of fabric, it has a probability of 95% of failing the contract. In any other transaction that the supplier is involved, it will have a probability of only 5% of failing the contract. In our experiments, we used five other types of behavior that represent five other different types of handicap: on a given fabric (*HFAB*), on low delivery times (*HDT*), on high quantities of a given fabric (*HFABQT*), on low delivery times for a given fabric (*HFABDT*), and on high quantities to be delivered in low delivery times (*HQTD*). As happen with the example before, a supplier has a probability of 95% of violating a contract if the current cfp matches the supplier’s handicap, and 95% of probability of succeeding the contract otherwise.

3.2 First set of experiments

3.2.1 Testbed and methodology

The first set of experiments was designed in order to evaluate the tendency of different trust models in exploring new business opportunities and how this capacity translates in terms of succeeded transactions. We used in this set of experiments three different models:

- *SA*: this model represents *SinAlpha*, a traditional trust aggregation system that uses properties of the dynamics of trust. The model was developed by us and is described in [6]. In there, we claim that this model gets better results than traditional statistical aggregation engines using weighted means;
- *CS*: this model goes one step further traditional trust models by considering contextual aspects of the business in assessment. It represents the model described in [11], a situation-aware technique that defines a context space as a n-dimensional metric space with one dimension per each represented situation feature. It is able to estimate trustworthiness values in unanticipated situations using the similarity between both situations. In the current experiments, we placed the reference contexts regularly over the combinations of possible values of the contractual attributes. This approach in a way represents situation-aware proposals that use domain specific, predefined similarity metrics to predict unanticipated situations ([12], [13], [14]);
- *CF*: this is the Contextual Fitness technique described in section 2.2 that is used in these experiments complimentary to the *SA* approach. As with the approach *CS* defined in the previous point, it is a situation-aware trust model. It was designed to fit well to non parochial open market scenarios, where the number of available trust evidences for a particular partner agent might be scarce.

3.2.2 Evaluation metrics

In this stage of experiments, we want to evaluate how client agents tend to behave in terms of selection of partners – and how good it is their decision on that – when using each one of the approaches defined above, representing respectively the traditional trust approach, more recent situation-aware trust models, and our own proposal for situation-aware trust assessment in scenarios that might involve scarcity of trust evidences.

Therefore, we use here two different metrics: the average utility got by all clients at every negotiation round, measured by the ratio given by the number of succeed contracts over the number of all contracts negotiated in the round; and the number of different suppliers that were selected by all the negotiating clients at every round.

3.2.3 Results

Figure 3 shows the results obtained in the first set of experiments. As can be observed from the graphic (bottom), both the SA and the CS approaches are relatively conservative concerning the selection of partners, where the 20 clients of the experiments choose in average between 9 and 10.5 different suppliers at each round.

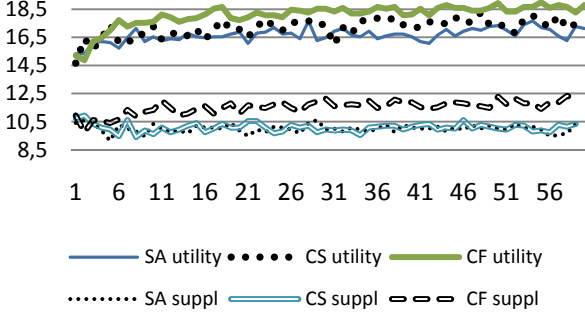


Figure 3. Average utility (top) obtained versus the average number of suppliers selected (bottom) at every negotiation round

On the other hand, the *CF* approach starts, since the first rounds, exploring a larger number of different suppliers and keeps showing this behavior all over the rounds. The described behavior of the approaches seems to be related with the utility that is achieved by them, as can be observed from the top plots of the graphic. In fact, the approach that is able to select from a greater number of different suppliers (the *CF* approach) also gets in average significant better utility (90.46%) than the other two approaches (83.30% for SA, and 85.87% for CS).

Another important result obtained with the *CF* approach is that, after some quick learning at the first rounds, the number of succeeded contracts with this method is very close to the maximum of 19 contracts that can succeed per round, i.e. to the 95% probabilistic limit imposed in the population of suppliers generated for our experiments.

3.2.4 Interpretation of results

At a first sight, we could expect that an approach that explores more business partners in the scenario described would have a smaller number of succeed contracts, at least in the first rounds of suppliers' exploration. However, the results show that the *CF* approach does not perform worse than the remaining representative approaches at this first exploration phase and performs significantly better than the others in the remaining steps of the experiments. This is due to the fact that the *CF* approach is able to extract tendencies of behavior with a reduced number of trust evidences.

In fact, the results obtained show how our approach is effective in discovering the particular characteristics of the population of agents in assessment, and how it is able to do so, irrespective of the number of trust evidences available for each agent under evaluation. Comparing to the traditional SA approach, *CF* reasons based not only on the global trustworthiness of the agent in assessment, but also on the context of the cfp for which the agent issued its proposal. On the contrary, the SA approach tends to select the agents that have the highest value of trustworthiness until that date. This parochial strategy results in the undesirable behavior of keeping choosing the same suppliers that occasionally fail the contracts for which they have a handicap for the current cfp, not giving a chance to other new suppliers.

On the other hand, the *CF* approach also outperforms the CS approach in the depicted scenario. Although CS is itself a situation-aware technique, it functions in a rather different way than *CF*. In

fact, when aggregating the trust evidences in order to compute the trustworthiness score of the agent in assessment, the CS model weights each trust evidence with the relative similarity between the evidence and the current cfp situation. When the number of trust evidences is scarce, it is not possible to populate all reference contexts that are sampled in the multi-dimensional evidence space, and the differences between different situations remain tenuous. In these conditions, the approach has a tendency to select, from the set of the more fitted suppliers, the ones that have already been involved in more contracts. Related to the SA approach, the CS approach has the benefit of contextualizing the decision, therefore achieving higher utility. But its relatively embedded parochial behavior explains why the results obtained are worse when compared to the *CF* approach.

3.3 Second set of experiments

3.3.1 Testbed and methodology

In the second set of experiments we generated two different populations of client agents in order to evaluate the potential benefits of choosing open market strategies, in terms of the global utility achieved by the clients. In these experiments, we used the *CF* approach, based on the results obtained in the first set of experiments that show that this approach is the more adequate to selecting partners in open market scenarios.

Therefore, we used the following client type of populations:

- *Parochial*: this population includes client agents that favor known trustworthiness suppliers instead of risking new, probably better suppliers. Their decision process of selecting partners is exclusively based on the trustworthiness scores of each supplier under assessment;
- *Non parochial*: this population includes client agents that seek to maximize their utility in the transacted operations. The decision process of these agents is based not only on the trustworthiness values of each supplier in assessment, but also in the expected *value* that come arise from the transaction with the supplier. In reality, these clients select the partner with whom they will trade based on their calculated *utility*, i.e. the *product* of the trustworthiness score of agents and their internal *value*, as explained below.

In this set of experiments, we introduced the notion of *value* of a contract. This *value* can be expressed through several characteristics of real world supplier connections, such as the price promised by the suppliers, the convenience of transacting with a given supplier, or even ethical and ecological concerns related to that particular partner, etc. In order to perform our experiments, each supplier has now an internal *value* that is assigned at the initialization time, following a uniform distribution on the set {0.50, 0.60, 0.70, 0.80, 0.90}.

Also, it is assumed that the value of an unknown supplier is 1.0, and the true value of the supplier is only presented to a given client after this client has already traded with the supplier. Through these settings, we expect that non parochial-based clients are motivated to explore new partners, as the potential calculated utility that arises from exploring outside the embedded space of relationships is high.

3.3.2 Evaluation metrics

In order to evaluate whether the exploration of new partners can increase the utility achieved by the clients, we compare both parochial and non parochial strategies based on the following different metrics: the *number of successful contracts* achieved by

all clients at every negotiation round and respective average number of contracts over all rounds; the *number of different suppliers* selected at every round; the *average utility* achieved by the clients at every round and its average score over all rounds.

3.3.3 Results

After performing the experiments, we observed that both approaches got similar results concerning the average number of successful contracts per client (parochial: 90.89%; non parochial: 90.67%). However, the non parochial strategy leads to a significant higher utility (75.25%) than the parochial strategy (68.59%).

Figure 4 plots the number of successful contracts, the number of different suppliers chosen and the utility observed per round, for each one of the two strategies. As expected, due to the use of the above introduced internal values (< 1.0), the achieved utility is always lower than the number of successful contracts per round.

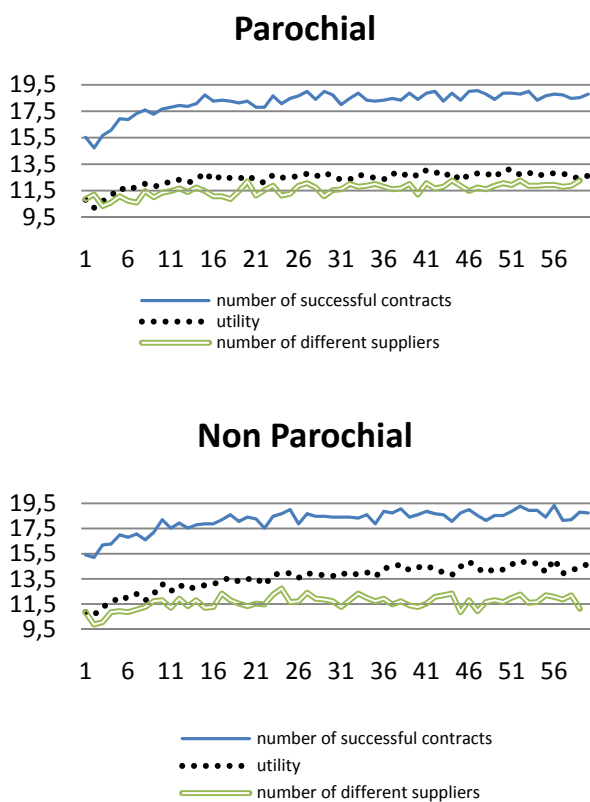


Figure 4. Comparison of the parochial and the non parochial clients' strategies

As can be observed in the figure above, the strategy used by the clients using the *CF* approach does not alter in a significant way the number of successful contracts achieved by the clients at every round or the number of different suppliers chosen per round. However, the big difference on the results obtained by each strategy resides on the utility achieved through them. In fact, after the first rounds of exploration, the non parochial strategy got systematically higher utility than the parochial strategy.

3.3.4 Interpretation of results

In this second set of experiments, we used the same trust method to support the process of partners' selection, which explains the similar results obtained by both strategies concerning the number of successful contracts and the number of different suppliers selected per round. This is due to the *CF* capability in

distinguishing between different handicapped partners and to reason based on their adequacy to the current business situation.

However, the most interesting conclusion extracted from this set of experiments is tied to the results obtained when the selection of partners is done taking into consideration not only the trustworthiness estimation of each supplier, but also the estimated value of each supplier, which mirrors a much more realistic situation. The results obtained have shown that the flexibility of the online tendency extraction of *CF* allows to safely exploring a larger space of opportunities, permitting the identification of different characteristics of suppliers. For instance, using the *CF* approach, a client agent feels safe to explore other potential trustable partners outside its previously known group of trustable suppliers, which in turn can bring extra utility (e.g. better prices) to the client business.

4 CONCLUSIONS

In this paper, we strongly support our previous firm belief that true open business-to-business markets need robust, computational assisted trust mechanisms that deal with heterogeneous, contextual and probably scarce evidences in order to effectively compute trustworthiness scores for business agents. However, current computational trust model proposals do not address these issues in a practical way. Therefore, we introduced in this paper a trust technique that is able to effectively extract tendencies of agents' behavior in different scenarios and contexts, even when the number of trust evidences is reduced. This technique can be used with any conventional trust aggregation engine.

Then, we experimentally evaluated how different trust model approaches, including our proposal, behave in environments where agents can seek business partners outside their breeding trading acquaintances. We verified that the focus on the online and incremental extraction of behavior features proposed by our technique effectively supports the exploration of new potential partners and, consequently, of new business opportunities, without jeopardizing the overall utility of business agents.

ACKNOWLEDGMENTS

The first author enjoys a PhD grant with reference SFRH/BD/39070/2007 from the Portuguese Fundação para a Ciência e a Tecnologia.

REFERENCES

- [1] R. Thurman, Purchasing contracts in the textile industry: how can purchasing contracts decrease subcontracting risks and reclamations in a textile industry company?, Tampereen ammattikorkeakoulu, Thesis Report (2007)
- [2] D. Stolle, In Trust in Society, ed. K. S. Cook (Russell Sage Foundation), 202-244 (2001)
- [3] M. W. Macy, Y. Sato, Trust, Cooperation, and Market Formation in the U.S. and Japan, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 99, No. 10, Suppl. 3, 7214-7220 (2002)
- [4] J. Sabater, M. Paolucci, On Representation and Aggregation of Social Evaluations. Computational Trust and Reputation Models, *Int. J. Approx. Reasoning* (2007)
- [5] C. M. Jonker, J. Treur, J., Formal Analysis of Models for the Dynamics of Trust Based on Experiences. In *Procs. of the 9th European Workshop on Modelling Autonomous Agents in Multi-Agent World: Multiagent System Engineering*. F. J. Garijo and M. Boman. LNCS, vol. 1647. Springer-Verlag, 221-231 (1999)
- [6] J. Urbano, A. P. Rocha, E. Oliveira, Computing Confidence Values: Does Trust Dynamics Matter?, in L. Seabra Lopes, N. Lau, P. Mariano & L. M. Rocha (eds.), *Progress in Artificial Intelligence (EPIA 2009)*, Springer, ISBN 978-3-642-04685-8, 520-531 (2009)
- [7] J. R. Quinlan, "Induction of Decision Trees", *Machine Learning*, 1(1),

1986, 81-106

- [8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, Volume 11, Issue 1, 2009
- [9] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993
- [10] J. Urbano, A. P. Rocha, E. Oliveira, Trustworthiness Tendency Incremental Extraction Using Information Gain, To appear in IEEE WI-IAT 2010, Toronto, Canada, (2010)
- [11] M. Rehak, M. Gregor, M. Pechoucek, Multidimensional context representations for situational trust. IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, 315-320 (2006)
- [12] A. Fabregues, J. Madrenas-Ciurana, SRM: a tool for supplier performance. AAMAS'2009, 1375-1376 (2009)
- [13] M. Tavakolifard, P. Herrmann, P. Ozturk, Analogical trust reasoning. Trust Management III, chapter 10, 149-163, (2009)
- [14] R. Hermoso, H. Billhardt, S. Ossowski, Dynamic evolution of role taxonomies through multidimensional clustering in multiagent organizations, Principles of Practice in Multi-Agent Systems, vol. 5925, chapter 45, 587-594, Springer Berlin Heidelberg (2009)