

# BEYOND TRUST. USING FUZZY CONTEXTUAL FILTERS FOR RELIABILITY ASSESSMENT IN MULTI AGENT SYSTEMS

**Esteve del Acebo Peña**

Dipòsit legal: Gi. 493-2014  
<http://hdl.handle.net/10803/132093>



<http://creativecommons.org/licenses/by/4.0/deed.ca>

Aquesta obra està subjecta a una llicència Creative Commons Reconeixement

Esta obra está bajo una licencia Creative Commons Reconocimiento

This work is licensed under a Creative Commons Attribution licence



Universitat de Girona

TESI DOCTORAL

---

BEYOND TRUST. USING FUZZY CONTEXTUAL FILTERS FOR RELIABILITY  
ASSESSMENT IN MULTI AGENT SYSTEMS

Esteve del Acebo Peña

---

2010









Tesi Doctoral

---

BEYOND TRUST. USING FUZZY CONTEXTUAL FILTERS FOR RELIABILITY  
ASSESSMENT IN MULTI AGENT SYSTEMS

Esteve del Acebo Peña

---

2010

Programa de Doctorat en  
Tecnologies de la Informació

Dirigida per:  
Prof. Josep Lluís de la Rosa

Memòria presentada per optar al títol de doctor per la Universitat de Girona



# Acknowledgments

It has been a long time since I first started the work that eventually would culminate in this thesis. In the meantime I have received the support and encouragement of too many people to enumerate them without fear to omissions. Nevertheless, I do not want to miss the opportunity to warmly express my gratitude to several people particularly. To Prof. Dominique Duhaut, who gently received me at the Laboratoire de Robotique de Paris during a bright and now distant parisian spring and called my attention upon the container transportation problem that was the origin of Bar Systems. To Prof. Antoni Ligeza with whom I published my first works about fuzzy rulebases quality criteria and who so kindly nursed me during my stay in Krakow. To Nicolás Hormazábal, who did all the Java programming work in the SPARTAN project. To my thesis advisor, Prof. Peplluís de la Rosa. To my parents, who are at least as happy as I am to see this work done. To Laura, who patiently endured the long process. Thanks.





# Publications

The main publications generated by the work presented in this thesis are the following:

- E. del Acebo and J.L. de la Rosa. A fuzzy system based approach to social modeling in multi-agent systems. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, Bologna, Italy*, pages 463–464, 2002.
- E. del Acebo and J.Ll. de la Rosa. Bar systems. a class of optimization algorithms for reactive multi-agent systems in real time environments. In *eNTeRTAI'n'06 Workshop: New Trends in Real-Time Artificial Intelligence. ECAI 2006*, 2006.
- E. del Acebo and J.Ll. de la Rosa. Introducing bar systems: A class of swarm intelligence optimization algorithms. In *Proceedings of the Swarm Intelligence Algorithms and Applications Symposium. SIAAS-08. Aberdeen, Scotland*, 2008.
- Esteve del Acebo, Nicolás Hormazábal, and Josep Lluís de la Rosa. Appraisal variance estimation in the art testbed using fuzzy corrective contextual filters. In *Artificial Intelligence Research and Development, Proceedings of the 10th International Conference of the ACIA, CCIA 2007, Sant Julià de Lòria, Andorra*, pages 104–111, 2007.
- Esteve del Acebo, Albert Oller, Josep Lluís de la Rosa, and Antoni Ligeza. Statistic criteria for fuzzy systems quality evaluation. In *Tasks and Methods in Applied Artificial Intelligence, proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE-98, Castellón, Spain, June 1-4, Volume II*, pages 877–887, 1998.
- Josep Lluís de la Rosa, Albert Trias, Antoni Martorano, Eloi Colomeda, David Huerva, and Esteve del Acebo. Shout and act. In *Proceedings*

*of the 2009 conference on Artificial Intelligence Research and Development: Proceedings of the 12th International Conference of the Catalan Association for Artificial Intelligence*, pages 91–100, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.

# Abstract

Trust modeling is widely recognized as an aspect of essential importance in the construction of agents and multi agent systems (MAS). As a consequence, several trust formalisms have been proposed over the last years. All of them have a limitation: they can determine the trustworthiness or untrustworthiness of the information received from a given agent, but they don't supply mechanisms for correcting this information, in the case of it being unaccurate, in order to extract some utility from it. In order to overcome this limitation, this thesis introduces the concept of reliability as a generalization of trust, and presents Fuzzy Contextual Filters (FCF) as reliability modeling methods loosely based on system identification and signal processing techniques. This thesis illustrates their applicability to two domains : the appraisal variance estimation problem in the Agent Reputation and Trust (ART) testbed and Bar Systems, a class of optimization algorithms for reactive MAS.



# Resum

El modelat de la confiança està àmpliament reconegut com un aspecte d'importància essencial en la construcció d'agents i de sistemes multi agent (MAS), i és com a conseqüència d'això que als darrers anys s'han proposat una nombrosa quantitat de formalismes de modelat. Tots ells però pateixen una limitació: poden determinar el grau de confiança que mereix la informació rebuda d'un altre agent, però no proporcionen mecanismes per a corregir aquesta informació, en cas de no ser acurada, amb vistes a extreure alguna utilitat d'ella. Per tal de superar aquesta limitació, aquesta tesi introdueix el concepte de fiabilitat (reliability) com a generalització de la confiança, i presenta els Filtres Difusos Contextuals (FCF) com a mètodes de modelat de la fiabilitat basats parcialment en tècniques d'identificació de sistemes i processament de senyals. La seva aplicabilitat es mostra en dos dominis: el problema de la estimació de la varianza de les taxacions a l'Agent Reputation and Trust (ART) testbed i als Bar Systems, una classe d'algorismes d'optimització per a MAS reactius.



# List of Figures

2.1	The set of filters of an agent act as a translative layer. . . . .	15
2.2	Structure of a fuzzy contextual filter . . . . .	17
2.3	Fuzzy sets for the example rule . . . . .	18
2.4	$2^X$ , the set of all the fuzzy subsets of $X = \{x_1, x_2\}$ . . . . .	20
2.5	Completeness of antecedent fuzzy sets . . . . .	24
2.6	Redundancy between a pair of fuzzy rules . . . . .	27
2.7	Consistency between a pair of fuzzy rules . . . . .	29
2.8	Error committed by ammeter $A$ for different values $K$ . . . . .	33
2.9	Representation of the problem . . . . .	34
2.10	Errors committed by the FCF . . . . .	36
2.11	Global reliability criteria values vs. errors committed by $A$ . . . . .	38
2.12	Global reliability criteria values vs. errors committed by $A$ . . . . .	39
2.13	Global reliability criteria values vs. errors committed by $A$ . . . . .	40
2.14	Consistency, redundancy and completeness vs. error . . . . .	47
2.15	Consistency, redundancy and completeness vs. error . . . . .	48
2.16	Consistency, redundancy and completeness vs. error . . . . .	49
2.17	Consistency, redundancy and completeness vs. error . . . . .	50
2.18	Consistency, redundancy and completeness vs. error . . . . .	51
2.19	Consistency, redundancy and completeness vs. error . . . . .	52
3.1	ART testbed logo. . . . .	58
3.2	The opinion transaction protocol . . . . .	60
3.3	Plot of the <i>conf</i> function . . . . .	67
3.4	Fuzzy sets corresponding to the rule . . . . .	72
3.5	Computation of the output of a FCF for ART . . . . .	73
3.6	SPARTAN <i>versus</i> IAM, Neil and Frost. . . . .	86
3.7	A new formula for the computation of the standard deviation of the relative error in appraisals. . . . .	89
4.1	Screen capture of the simulation program in action. . . . .	104
4.2	Time needed by the system to deliver all the containers . . . . .	105



4.3	Time needed by the system to deliver all the containers . . .	105
5.1	The graphical interface to the simulation . . . . .	112
5.2	Fuzzy sets for variables <i>distance</i> , <i>profit</i> and <i>interest</i> . . . . .	113
5.3	The matrices representing the fuzzy systems used by each agent type . . . . .	117
5.4	Performance increment in agents using evolving FCFs . . . . .	119

# List of Tables

3.1	Final results of the experiments in the ART testbed. . . . .	86
-----	--	----



# List of Algorithms

2.1	Confidence computation function . . . . .	45
3.1	General appraisal algorithm for the SPARTAN agent. . . . .	65
3.2	One iteration of SPARTAN in the ART testbed. . . . .	85
4.1	Individual agents' behavior algorithm in Bar Systems . . . . .	98
5.1	Main procedure of the simulator . . . . .	111
5.2	Simple evolutionary algorithm . . . . .	116



# Table of Contents

<b>Acknowledgments</b>	v
<b>Publications</b>	vii
<b>Abstract</b>	ix
<b>Resum</b>	xi
<b>List of Figures</b>	xi
<b>List of Tables</b>	xiv
<b>List of Algorithms</b>	xv
<b>Introduction</b>	1
<b>I Fuzzy Contextual Filters</b>	<b>5</b>
<b>1 Social Intelligence and Trust</b>	<b>7</b>
1.1 Artificial Social Intelligence . . . . .	7
1.2 The Need for Trust Formalization in MAS . . . . .	10
1.3 Trust Formalisms . . . . .	11
<b>2 Fuzzy Contextual Filters</b>	<b>13</b>
2.1 Beyond Trust. Reliability . . . . .	13
2.2 Fuzzy Contextual Filters . . . . .	14
2.3 The Corrective Module . . . . .	16
2.4 The Reliability Calculation Module . . . . .	19
2.4.1 The Sets-as-Points Framework. . . . .	20
2.4.2 The Subsethood Relationship . . . . .	21

2.4.3	Three structural criteria for fuzzy systems reliability assessment . . . . .	22
2.4.3.1	Notation . . . . .	22
2.4.3.2	Completeness . . . . .	23
2.4.3.3	Redundancy . . . . .	25
2.4.3.4	Consistency . . . . .	28
2.4.4	The reliability function . . . . .	30
2.5	Case Example: Calibration of an Ammeter . . . . .	31
2.5.1	Modeling the ammeter . . . . .	31
2.5.2	Building the Fuzzy Contextual Filter . . . . .	32
2.5.3	The reliability calculation module . . . . .	35
2.5.4	Results . . . . .	35
2.6	Conclusions . . . . .	46
<b>II</b>	<b>Applications</b>	<b>55</b>
<b>3</b>	<b>Using FCFs for Variance Estimation in the ART Testbed</b>	<b>57</b>
3.1	Introduction . . . . .	57
3.2	The ART Testbed . . . . .	57
3.2.1	Client Appraisals . . . . .	58
3.2.2	Opinion Transactions . . . . .	59
3.2.3	Reputation in the ART Testbed . . . . .	61
3.2.4	Client Share Assignment . . . . .	61
3.3	Several Considerations on the Client Share Assignment Function	62
3.4	The SPARTAN Algorithm . . . . .	64
3.5	The Taxation Vectors . . . . .	66
3.6	Translating Standard Deviation of Relative Errors to Confidence Values . . . . .	66
3.6.1	Translating Confidence Values to Standard Deviations	67
3.7	Determining the Optimal Set of Weights in Appraisal Combination . . . . .	68
3.8	Variance Estimation Using Fuzzy Contextual Filters . . . . .	70
3.8.1	The Reliability Calculation Module . . . . .	74
3.9	Deciding How Much Spend in Creating each Appraisal . . . . .	75
3.9.1	Finding a Good Value for $c_g$ . . . . .	75
3.9.2	Deciding Which Appraisals to Buy . . . . .	80
3.10	Results . . . . .	84
3.11	Conclusions and Proposals to the ART Testbed Designers . . . . .	87

<b>4</b>	<b>From Swarm Intelligence to Crew Intelligence: Bar Systems</b>	<b>91</b>
4.1	Introduction . . . . .	91
4.2	Swarm Intelligence . . . . .	92
4.3	Swarm Intelligence vs. Crew Intelligence . . . . .	93
4.4	Bar Systems . . . . .	94
4.4.1	Definition . . . . .	95
4.4.1.1	The $cost(a_i, t_j)$ function . . . . .	96
4.4.2	Specialization in Bar Systems . . . . .	97
4.4.3	Bar System's Dynamics . . . . .	97
4.4.4	Inter-agent communication in Bar Systems . . . . .	97
4.4.4.1	Negotiation in Bar Systems . . . . .	99
4.4.5	Local planning in Bar Systems . . . . .	99
4.5	The CONTS Problem . . . . .	100
4.5.1	Definition of the problem . . . . .	100
4.5.2	Complexity of the CONTS problem . . . . .	101
4.6	A Bar System for Solving the CONTS Problem . . . . .	101
4.6.1	Inter-agent communication and local planning for the CONTS problem . . . . .	103
4.7	Results . . . . .	103
4.8	Conclusions and Future Work . . . . .	107
<b>5</b>	<b>Using FCFs to Filter Inter-Agent Communication in Bar Sys- tems</b>	<b>109</b>
5.1	Introduction . . . . .	109
5.2	Case Study. Competitive taxi driver agents . . . . .	110
5.2.1	The problem . . . . .	110
5.2.2	The simulation . . . . .	111
5.2.3	The fit function. Computing the preferred customer . . . . .	112
5.2.4	Filtering the fit values provided by the agents . . . . .	114
5.2.5	A simple evolutionary algorithm . . . . .	115
5.3	Results . . . . .	116
5.4	Conclusions and Future Work . . . . .	118
<b>6</b>	<b>Conclusions</b>	<b>121</b>
<b>III</b>	<b>Appendices</b>	<b>123</b>
<b>A</b>	<b>Fuzzy Sets and Fuzzy Systems</b>	<b>125</b>
A.1	Definitions on Fuzzy Sets . . . . .	125
A.2	Fuzzy Systems . . . . .	126



<b>B</b>	<b>Ammeter Calibration Case Example Results</b>	<b>129</b>
B.1	Ammeter Calibration Case Example Results . . . . .	129
<b>C</b>	<b>The Half Normal Distribution</b>	<b>147</b>
C.1	The Half Normal Distribution . . . . .	147
<b>D</b>	<b>The Complexity of the CONTS Problem</b>	<b>151</b>
D.1	CONTS és NP-Hard . . . . .	151
D.1.1	Problemes de Decisió i Classes de Complexitat . . . . .	154
D.1.2	El Problema del Viatjant de Comerç (TSP) . . . . .	156
D.1.3	DTSP* és NP-complert . . . . .	157
D.1.4	DCONTS és NP-complert . . . . .	158
<b>E</b>	<b>SPARTAN in the Second ART Testbed Competition</b>	<b>159</b>
E.1	SPARTAN in the Second ART Testbed Competition . . . . .	159
	<b>Bibliography</b>	<b>162</b>

# Introduction

## Motivation

Agents, the new emerging trend in Artificial Intelligence, are

Computer systems capable of flexible autonomous action in dynamic, unpredictable and open environments endowed with the capacity to interact with other systems (artificial or natural). [51]

Agents are, then, situated entities. That is, they are embedded in an environment and continually operate into it in a three steps loop: perceive the environment, decide what action to take and act over the environment consequently. Agents situated in the physical world are known as physically situated agents.

When two or more agents are situated in the same environment and interact with each other in a coordinated manner, they form a multi-agent system (MAS). In MAS, individual agents typically pursue its own agenda of goals, whose fulfillment may require the interaction with other agents within the environment. So, agents in a MAS exist in a social environment determined by:

- The set of agents in the environment, as well as the set of relationships established between them.
- A culture, understood as a corpus of common knowledge and a set of norms regulating the social behavior of the agents.

Agents operating in a social environment are called socially situated agents. A consequence of the social situation is the necessity of a set of social skills which allow agents to be able to perform efficiently within its social environment. This set of skills include, among others, the ability to communicate, cooperate or negotiate with other agents and constitute the artificial social intelligence of the agent.

One of the most important social skills for an agent is the ability to decide when and to what extent trust another agent. It is clear that some amount of trust is necessary if the agent has to collaborate or coordinate himself with other agents (the same thing can be said about we humans, we could hardly risk to cross a street at the crosswalk without some degree of trust in the drivers waiting for the light to go green), but it is also clear that, in order to avoid exploitation from other agents, trust can not be unconditional.

Social situated agents, then, must have at their disposal some kind of trust assigning method, which has to take into account two distinctive features of trust:

- Trust is dynamic. Trustworthiness of a given agent can change over time. Any trust model has to take into account the history of interactions, and agents ought to be able to modify the model upon subsequent interactions.
- Trust is contextual. An agent can be very trustworthy given advice about cooking but very untrustworthy given advice about playing bridge. Similarly, an agent can be trustworthy only to a class of agents (good customers, for example). Finally, an agent's trustworthiness can depend on other external facts such as room temperature or the amount and kind of the other agents making requests to him.

Over the last years, several attempts to devise trust formalizations and trust assignment methods for MAS have been proposed from diverse points of view (recommender systems, social networks, electronic commerce. . .). An almost common trait is that they just provide an agent with some kind of value (let it be numeric, boolean, a category or even a fuzzy set) somehow representing the trust that can be assigned to other agents in a given circumstance. Therefore, their usefulness is limited just to help an agent to decide, in a given circumstance, whether to believe another agent or not.

The main point of the present PhD. is: in an interaction between agents consisting of an information exchange (and pretty much every interaction between agents can be seen as an information exchange), an agent can extract some utility from the information provided by another agent even if the first agent does not trust the second agent (that is, believes that the information provided by the second agent is probably false). An exemple may help to make this point clear: imagine a broken TV remote control that makes the TV display Discovery Channel whenever the button for Eurosport is pressed, display ShopTV when we press the Euronews button and so on, in such a way that no button is correctly associated to its corresponding channel. Obviously you can not trust the remote control, but that does not mean you ought to throw it away. On the contrary, you only need to learn the

new button-to-channel correspondences (or repaint the top of the buttons) to have a perfectly useful remote control. As another example, imagine that some measuring device tends to give bad measurements as room temperature increases. Clearly we can not trust the readings of the device in high temperature conditions, they will be erroneous, but if we manage to figure out how the error committed by the device depends upon the temperature, given a reading and a temperature we would be able to subtract the error and obtain a more or less accurate result.

The aim of this Ph.D is to devise a capture mechanism for the patterns and regularities found in the erroneous or false information received by an agent in a MAS in such a way that the agent could, in subsequent interactions, filter or correct the information received by that same agent in order to make it useful. We argue that such a mechanism will enhance the social intelligence of the agents, reducing the amount of communication errors and thus making the whole MAS more robust.

## Overview and main contributions

This work proposes a shift from the concept of trust to the concept of reliability. That is: does the agent respond consistently to its inputs? Does it tend to give the same or a similar answer under the same or similar stimuli? If so, after a number of interactions it will be possible to build some kind of filter or translator able to eliminate the error and produce useful information.

In order to prove that claim, Fuzzy Contextual Filters (FCFs) are introduced. A FCF has, as input, the information received from an agent, jointly with as many other variables as necessary to specify the context in which the interaction has taken place, and produces, as output, a corrected version of the information received (corrected in the sense that the FCF tries to eliminate the errors contained in it) as well as a value (the reliability value) stating how much confidence it has in the correctness of the filtered information. The corrective module of a FCF consists mainly of a fuzzy rule base which codifies, in the form of fuzzy rules, all the past interactions between the agents. New interactions add new rules to the rulebase. The reliability calculation module computes the reliability value by examining the structure of the fuzzy rule base and applying three new criteria for fuzzy rule bases quality assesment: completeness, consistency and redundancy. Those criteria are generalizations of their crisp counterparts based in Kosko's Fuzzy Subsethood Theorem [41].

The second part of the thesis illustrates the applicability of FCFs to two different domains, the Agent Reputation and Trust (ART) testbed and Bar Systems. The ART testbed was devised as a framework for the implementation and testing of trust and reputation formalisms. ART championships

are held regularly and a FCF-equipped agent, SPARTAN, participated in the Second International ART competition, held in conjunction with the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007). It qualified fifth in the preliminary round and then won a position in the final round to finish in fourth place.

Bar Systems, also a main contribution of this thesis, are a family of very simple algorithms for different classes of complex optimization problems in static and dynamic environments by means of reactive multi agent systems. Bar Systems belong to the family of Crew Intelligence algorithms, an extension of Swarm Intelligence algorithms that endows individual agents with additional communicative and local planning abilities. Bar Systems are loosely inspired in the behavior that a crew of bartenders can show while serving drinks to a crowd of customers in a bar or pub. This thesis will show how Bar Systems can be applied to a NP-hard scheduling problem, and how they achieve much better results than other greedy algorithms in the “nearest neighbor” style. It will also show how the use of FCFs can enhance the performance of an agent using a Bar Systems algorithm in a competitive MAS.

## **Structure of the document**

This thesis is structured in three parts. In part I (Fuzzy Contextual Filters), chapter 1 discusses social intelligence, trust and the need for trust formalisms in MAS. Chapter 2 presents the concept of reliability as an extension of trust and introduces Fuzzy Contextual Filters. It also includes a case study to illustrate the applicability of FCFs. In part II (Applications), chapter 3 shows the application of FCFs to the variance estimation problem in the ART testbed and presents several techniques and algorithms used by the SPARTAN agent. Chapter 4 introduces Bar Systems and shows how they can be used to obtain good approximated solutions to a NP-hard scheduling problem. Finally chapter 5 presents a competitive MAS (taxis competing for customers) and show how agents using FCFs to filter the (probably intentionally erroneous) information provided by other competitor agents can increase its performance. Part III contains several appendices.

Part I

# Fuzzy Contextual Filters



# Chapter 1

## Social Intelligence and Trust

### 1.1 Artificial Social Intelligence

The wide ensemble of abilities that allows humans to, among other things, reason, learn, communicate with each other, deal with new situations and apply knowledge to manipulate our environment, which is called collectively intelligence, is a multiple faceted phenomenon. Edward L. Thorndike gave to this notion the shape of a scientific theory as early as 1920 [48, 55], when he drew an important distinction among three broad classes of intellectual functioning: *abstract intelligence* (the one measured by standard intelligence tests), *mechanical intelligence* (the ability to visualize relationships among objects and understand how the physical world works) and *social intelligence* (the ability to function successfully in interpersonal situations). In spite of this, historically, the bulk of the research effort made by both Psychology and Artificial Intelligence communities has headed towards the study of the abstract, classical, part of the intelligence, to the point of most authors reducing social intelligence just to general intelligence applied to social situations. The reason for this can, perhaps, be found in the lack of adequate instruments (in the style of IQ tests) for the measurement of the less conventional aspects of intelligence and, on the other hand, in the relative success of early AI research in the development of modeling mechanisms for classes of tasks directly related to abstract intelligence (i.e., reasoning, planning and problem solving).

However, this situation has changed over the last years. In Psychology, the appearance of the *Machiavellian Intelligence Hypothesis* [9, 42], according to which primate intelligence originally evolved to solve social problems and was only later extended to problems outside the social domain, has rapidly increased the interest on the study of social aspects of intelligence. A similar phenomenon has happened in the AI field after the shift to the



agent paradigm. The agent paradigm contemplates the physical situation of agents, the tight coupling between the agent and its environment, as unavoidable requirement in order to build intelligent agents<sup>1</sup>. An agent's environment contains typically other agents with whom it has to interact. This defines a social environment and justifies the necessity of situation from the social point of view. Following Edmonds [27]:

“In a physical situation the internal models may be insufficient because of the enormous computation capacity, amount of information and speed that would be required by an agent attempting to explicitly model its environment. In a social situation, although the speed is not so critical, the complexity of that environment can be overwhelming and there is also the obvious external computational resources provided by the other agents and their interactions. This means that an agent can be said to be socially situated by analogy with being physically situated - in both cases the balance of advantage lies in using external causal processes and representations rather than internal ones.”

A consequence of the embedding of the agent into the social environment is the necessity of development of a set of skills which allow the agent to perform efficiently within it. As usual, there is not universal agreement about the precise meaning of Social Intelligence (SI) and Artificial Social Intelligence (ASI) (in fact, Edmonds remarks in [31] that the term social intelligence is ambiguous in the sense that it can either indicate the intelligence that an individual needs to effectively participate in a society, or the intelligence that a society as a whole can exhibit). Duffy [25] defines social intelligence as “the intelligence that underlies behind group interactions and behaviours” while Cantor and Kihlstrom [39] redefine the term to refer to “the individual’s fund of knowledge about the social world”. Edmonds [26] proposes the Turing Test as a criteria for determining the achievement of truly social intelligence while Hoggs and Jennings [37] prefer to talk about social rationality<sup>2</sup>. Kerstin Dautenhahn [14], finally, gives perhaps the most cited definition of social intelligence as:

---

<sup>1</sup>See, for example, in [31], Franklin and Graesser’s definition of agent as “a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future”.

<sup>2</sup>Extending Newell’s *Principle of Rationality* to state the *Principle of Social Rationality* : “If a socially rational agent can perform an action whose joint benefit [for the whole society] is greater than its joint loss, then it may select that action.”

“the individual’s capability to develop and manage relationships between individualized, autobiographic agents which, by means of communication, build up shared social interaction structures which help to integrate and manage the individual’s basic (“self-ish”) interests in relationship to the interests of the social system at the next higher level. The term artificial social intelligence is then an instantiation of social intelligence in artifacts.”

Neither is there a general agreement upon the way ASI has to be implemented or even what its final goal has to be. Researchers coming from “classic” AI mostly focus in the human-agent social interaction (i.e., the “human in the loop” approach [16]). From this point of view ASI has to serve a double purpose: in one hand, to facilitate the interaction between agents and humans and, on the other hand, the study of human social processes through the development of suitable social models. The architecture of this type of social agents uses to be a extension of some form of BDI architecture (e.g., [46]) and its design tends to follow the *Life-Like Agents Hypothesis*<sup>3</sup> [15]. Two examples of this type of social agents are the AURORA Project [14], a remedial tool for getting children with autism interested in coordinated and synchronized interactions with the environment and the *Let’s Talk!* socially intelligent agents for language conversation training [49].

On the other side, research coming from the social sciences community is focused in social simulation. That is, the design of synthetic societies of agents in physical or virtual environments in order to study the emergence and evolution of social phenomena like cooperation, competition, trust, reputation, markets, social networks dynamics, norms and languages. The interested reader can find a classical introduction to the field in [35]. The significance of this aspect of ASI has to be expected only to increase due to the gaining importance that electronic markets and virtual societies will have in the years to come.

Finally, a third aspect of ASI research has its roots arguably in Artificial Life and Distributed Problem Solving. It is the “Engineering with Social Metaphors” approach to ASI, which tries to devise socially inspired problem solving techniques and algorithms. Perhaps the most representative class of such techniques are those based in Swarm Intelligence [7].

Whichever point of view is chosen, the field of Socially Intelligent Agents is a fast growing and increasingly important area that comprises highly active

---

<sup>3</sup> “Artificial social agents (robotic or software) which are supposed to interact with humans are most successfully designed by imitating life, i.e. making the agents mimic as closely as possible animals, in particular humans.”

research activities and strongly interdisciplinary approaches coming from as diverse fields as Organizational Science, Philosophy, Cognitive Science, Artificial Intelligence, Cybernetics and Social Simulation.

## 1.2 The Need for Trust Formalization in MAS

Trust is one of the main concepts upon which human and animal societies are built. It is evident, therefore, the importance of its formalization for the construction of artificial or electronic societies, which so vast amount of interest have caused not only in the Artificial Intelligence and Computer Science communities, but also in such different ones as Sociology, Economics and Biology. Quoting [51]:

“Artificial Intelligence is quickly moving from the paradigm of an isolated and non-situated intelligence to the paradigm of situated, social and collective intelligence. This new paradigm of the so called intelligent or adaptive agents and Multi-Agent Systems (MAS), together with the spectacular emergence of the information society technologies (specially reflected by the popularization of electronic commerce) are responsible for the increasing interest on trust and reputation mechanisms applied to electronic societies.”

By definition, an essential characteristic of MAS is the existence of an information interchange between the individual agents forming the system. In the case of collaborative MAS, the aim of this communication is the improvement of the global performance of the system. Therefore agents, in general, do not lie each other consciously. In the case of competitive environments, however, individual agents are selfish, in the sense that its behavior is addressed to maximize some kind of individual utility function, even if that means a prejudice for the individual interests of the other agents or the diminution of the overall performance of the system. Communicative acts in competitive MAS are therefore addressed to obtain individual benefit and it is more suitable (because it can be profitable) the conscious communication of false information.

Both in collaborative and in competitive MAS, however, an emitter agent can communicate false information to a recipient agent because of several reasons. The main ones being:

1. The emitter agent is, simply, wrong. He is honest, in the sense that he believes he is communicating a true statement, but the transmitted information is false.

2. Emitter and recipient agents do not use the same language. The message encloses a true statement, as understood for the emitter agent, but has a different and false meaning for the recipient agent. That's why ontologies are used, just to try to assure that all the agents in a domain speak the same language
3. A transmission error occurred. The emitted and received messages are different.
4. The emitter agent consciously transmits a false information to the recipient agent. The aim of such behavior can be supposed to be the obtaining of some benefit from the prejudicing of the recipient agent. That is the typical behavior we can expect in competitive environments

Whatever could be the reason behind the transmission of false information, individual agents need some kind of mechanism that allow them to deal with it. Agents can't afford (specially in competitive environments) to believe everything the other agents tell to them. A car vendor agent who commits itself to deliver a car "soon" and who says that the car is "fast" can be honest even if the car lasts a year to arrive and it can not run faster than 100 kilometers per hour. Perhaps he really believed what he was saying, perhaps the words "soon" and "fast" have a different meaning in the car vending language or even, perhaps, he said "late" and "slow" but somehow the sounds changed in their way from their mouth to our hears. More probably, however, he is deliberately lying to take profit from us. In either case, we need to learn from our experience in order to know what can be expected from him in further deals. Here is where trust and reputation modeling methods come in as an important field of study inside the theory of MAS.

### 1.3 Trust Formalisms

Over the last years, several attempts to devise formalization for the concepts of trust and reputation have been carried out from diverse points of view (recommender systems, social networks, electronic commerce...). In his PhD dissertation [51], Sabater proposes a set of traits that can allow to characterize and classify them: paradigm type (cognitive or numerical), the information sources used to build the model (direct experiences, witnesses information, sociological information and prejudice), visibility types (subjective and global), model's granularity (context dependent or non context dependent), agent behavior assumptions (basically whether the formalism includes specific mechanisms to deal with liars), model type (trust and reputation) and type of exchanged information (boolean and continuous).

Amongst the most widely recognized trust and reputation formalism we can enumerate those of Marsh [44], the Sporas and Histos models [60], the

models proposed by Schillo et al. [53], Abdul-Rahman and Hailes [1], Esfandiary and Chandrasekharan [29], Yu and Singh [58, 59], Sen and Saija [54], Carter et al. [11], Castelfranchi and Falcone [12] and the AFRAS [10] and REGRET [51] models. Large world wide web marketplaces like Amazon and Ebay have also developed their own trust modelling frameworks.

It is beyond the scope of this dissertation to discuss the details and merits of each of those formalisms. A detailed review of each of them can be found in Sabater's PhD. dissertation [51], along with a comparison attending the classification dimensions enumerated below.

## Chapter 2

# Fuzzy Contextual Filters

### 2.1 Beyond Trust. Reliability

While the the trust formalisms briefly enumerated in Chapter 1 can provide a number, category or even fuzzy statement measuring the trustworthiness of a given agent or, more precisely, the trustworthiness of the information provided by a given agent, they fall short, in our opinion, in the sense that they don't supply any filtering or correcting method in order to make the provided information useful, even if wrong. That is, they seem to oversee the fact that, in some cases, false information transmitted by an agent can be useful, if conveniently filtered.

*It is not necessary to trust an agent (in the sense of believing it is saying the truth) in order to get some utility from the information provided by it. This information can be useful even if it is false, if it exists some method able to correct it.*

Let's put an example: a watch agent that goes two and a half hours in advance will never tell you the right time, so you will do good not trusting it. Does it implies that you can't get any utility from it?. Quite on the contrary, you can completely rely on it. Its regularity makes possible to correct the information it provides and get the exact time, a thing that would be impossible to do accurately with a watch that goes only one minute in advance half the time and one minute in retard the other half, at random. We can say much the same thing about our car vendor agent. Better for us to do not believe everything he could tell us, of course, but even if we don't trust him, we can yet extract some probably useful information from his offers, perhaps in the form of upper or lower bounds. Moreover, with the time, if we deal with him often enough, we can arrive to learn its language, that is, to capture

regularities in it which can allow us to, for example, reject at once a car if he says of it to be “not very old”.

The key concept in order to be able to correct messages coming from other agents is *reliability*<sup>1</sup>. *If an agent tends to communicate similar information under similar circumstances, a moment will arrive when we will be able to extrapolate the circumstances, more or less correctly, from the received messages.* On the contrary, if an agent emits just random messages it will be very difficult, if not impossible, to obtain from them any utility at all.

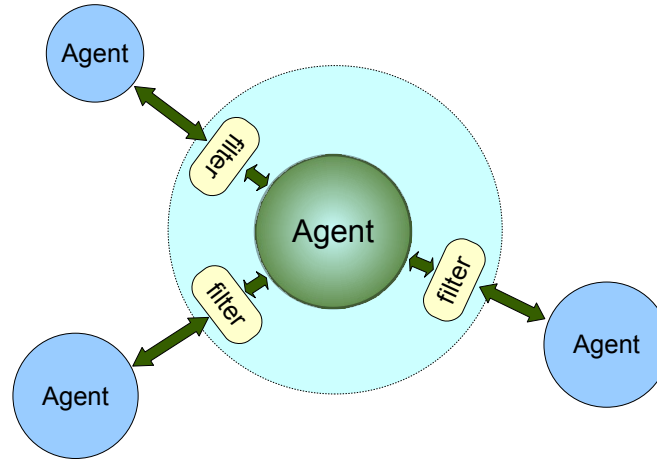
These corrective mechanisms able to convert false but reliable input information into correct information, which we will call filters, can have very different structures. The filter for a watch agent that goes two hours and a half in retard could be as simple as adding 150 minutes to the time he says it is. On the other hand, we will need a much more sophisticated filter when dealing with the car vendor agent, maybe some kind of expert system. In the following section we will show how simple filters, based on fuzzy systems, can be constructed and how they can be learned and used to improve the performance of individual agents in their social environment.

## 2.2 Fuzzy Contextual Filters

Think about the following problem: An agent  $A$  interacts with several other agents in a multi-agent environment requesting from them some kind of information, which they supply (this information can be false because of any of the reasons exposed in Section 1.2). Suppose also that the correct answers to  $A$ 's requests are made available to  $A$  by the environment in a posterior time instant, in such a way that  $A$  is able to know which agents told the truth and which agents lied, and how much. Our point is: for  $A$  to be able to perform well in this kind of environment it has to maintain a set of filters (one of them for each agent it interacts with) which allows it to correct the information received from the other agents, as well as to assess the possible utility of the corrected information. These filters must be dynamic, in the sense that they must evolve and adapt to changes in the environment and in the behavior of the other agents. So, (see figure 2.1) filters will act as a translative layer easing the process of interpretation of the messages sent by other agents. They can also, on the other hand, help the agent to translate the information it wants to transmit to the language spoken by the other agents, increasing therefore the probability of being correctly understood.

---

<sup>1</sup>From reliable, in the sense of “giving the same result in successive trials”. (From the *Merriam-Webster Online Dictionary* [45])



**Figure 2.1:** *The set of filters of an agent act as a translative layer.*

As we have said, it is also very important for the agent that owns the filter to have some kind of measure of the correctness of the filtered information, that is, the degree to which it can be expected to reflect the reality. We will call this value *reliability* and the filter will compute it basically from the observed regularities in the behavior of the filtered agent in past interactions.

Figure 2.2 shows the suggested structure for the construction of these filters, which we call fuzzy contextual filters (FCFs) [19]. A FCF  $F$  has two parts, the *corrective module* and the *reliability calculation module*. The corrective module is in charge of converting the probably false input information or data provided by the agents into correct useful information that reflects better the reality. The reliability calculation module tries to measure the confidence the agent can have in the correction of the filtered information provided by the corrective module. Obviously, the reliability value has to be high for predictable agents (those that act in a similar way under similar circumstances) and low for the unforeseeable, chaotic or random ones. In the following sections we will take a closer look to those two modules.



## 2.3 The Corrective Module

The corrective module is a special case of a Mamdani fuzzy inference system<sup>2</sup> where the fuzzy rules have the form:

If  $A_1$  is  $S_1$  and ... and  $A_n$  is  $S_n$  and  $V$  is  $L_1$  then  $W$  is  $L_2$

where:

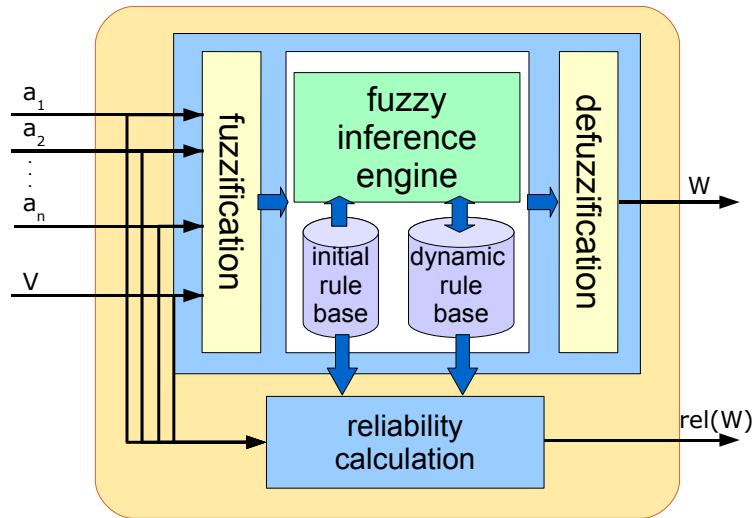
- $S_1, S_2 \dots S_n$  are linguistic labels, defined by fuzzy sets on universes of discourse  $X_1, X_2 \dots X_n$ , respectively.
- $A_1, A_2 \dots A_n$  are fuzzy variables taking values over the fuzzy power sets of  $X_1, X_2 \dots X_n$ , respectively.
- $L_1$  and  $L_2$  are linguistic labels defined by fuzzy sets over the universes of discourse  $U_1$  and  $U_2$ , respectively.  $U_1$  and  $U_2$  can be, and usually are, the same set.
- $V$  and  $W$  are fuzzy variables taking values over the fuzzy power sets of  $U_1$  and  $U_2$ , respectively.

We will call  $A_1, A_2 \dots A_n$  the *context variables*,  $V$  the *main variable* and  $W$  the *filtered variable*. We can see the operation of the corrective module as a transformation of fuzzy sets over the universe  $U_1$  to fuzzy sets over the universe  $U_2$  (which will be usually the same as  $U_1$ ) in a way that depends on the values of the context variables as well as on the value of the main variable. The corrective module of a FCF, then, filter the values (fuzzy sets) of the main variable to obtain new values (fuzzy sets over the same universe or another one) which are expected to be more suitable for some purpose. As is the case with general Mamdani fuzzy systems, it is possible to use FCFs on crisp input values to produce crisp filtered values by using appropriate fuzzification and defuzzification procedures.

The rule base of the corrective module has two components, the static and dynamic rule bases. The static rule base is fixed (and possibly the same) for every agent. It expresses the *a priori* assumptions about the behavior of the other agents in the environment and serve as a departing point in the interpretation of other agents's assertions. It can be as simple as the identity function or can, for instance, incorporate some common sense knowledge about the behavior which can be expected from certain kinds of agents. The

---

<sup>2</sup>A explanation of fuzzy sets, fuzzy logic and fuzzy inference systems' theories is beyond the scope of this thesis. The reader can find several basic definitions in Appendix A.



**Figure 2.2:** Structure of a fuzzy contextual filter

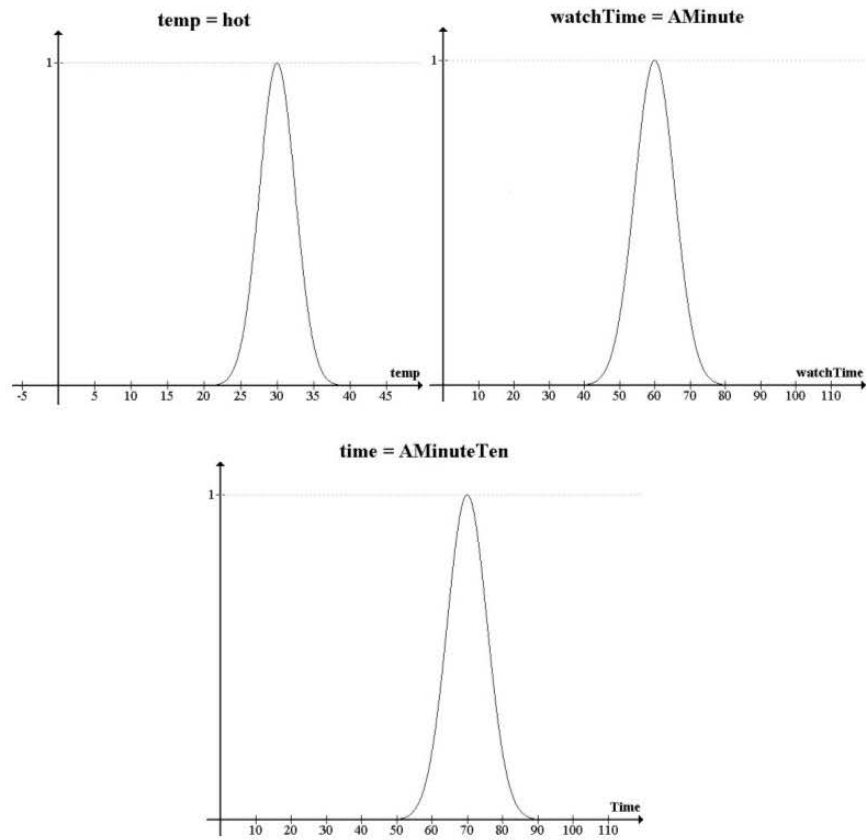
purpose of the initial rule base is to provide the system with a sensible initial structure from which the filtering process can start. The influence of the initial rule base in the general filtering process must diminish as new rules, due to direct interactions with the filtered agent, are incorporated into the dynamic rule base.

The dynamic rule base is built upon the information extracted (in the form of fuzzy rules) from the interactions between the agent which owns the filter and the filtered agent. Each interaction involves the observation of several values at a given time  $t$ : the values of the contextual variables, the predicted value for the output value and the true observed value. Returning to the watch example introduced in section 2.1, think about a stopwatch whose exactness depend on the temperature and assume that it tends to run slow with lower temperatures and to gain with higher ones. One interaction with the stopwatch agent will involve the observation at a given time of the value of the temperature, the time marked by the stopwatch and the true, exact time. Such an interaction can allow the FCF to build a fuzzy rule of the form:

If *temp* is *Hot* and *watchTime* is *AMinute* then *time* is *AMinuteTen*

for suitable fuzzy sets defining the labels *High*, *AMinute* and *AMinuteTen* (see Figure 2.3).

A new rule can be added at each interaction, so the rule base is dynamic in the sense that it evolves with time and can adapt itself to changes in the



**Figure 2.3:** Fuzzy sets for the example rule: *If temp is Hot and watchTime is AMinute then time is AMinuteTen*

environment and in the behavior of the filtered agents.

So, the construction of the dynamic rule base can be viewed as a system identification task where the behavior of the filtered agent has to be modeled from a set of examples, the results of past interactions between the modeling and the modeled agents. As a system identification problem, several modeling methods can be used, ranging from those based on a neuro-fuzzy, backpropagation-like approach (Jang's ANFIS [38] would be a good example of this) to those based on lookup tables [57] or, even, genetic algorithms [13]. We will see an example of a somewhat simpler approach in the case example of section 2.5.

## 2.4 The Reliability Calculation Module

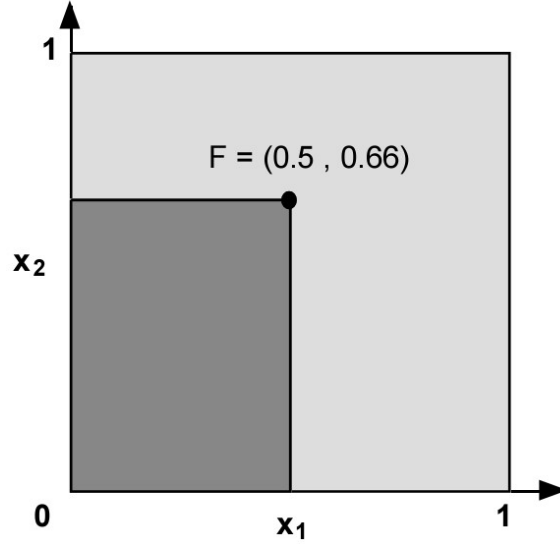
The function of the second part of the FCF, the reliability calculation module, is to compute the reliability of the value of the filtered variable obtained by the corrective module. Reliability is a function of the main and context variables, that is, given values for the main and context variables, the reliability calculation module must produce a value representing the confidence we can have in the exactitude of the value of the filtered variable computed by the corrective module. Reliability will depend upon the number of prior similar interactions between filtering and filtered agents as well as upon the regularities observed during that interactions.

The problem can be stated as follows: given a fuzzy system defined by a set of fuzzy rules obtained from a set of interactions like the one explained in the last section, how can we define a set of criteria that allow us to assess the reliability of the fuzzy system output for given values of the input variables?

There are different classes of such criteria. On one hand we have performance based criteria, where confidence is given to the systems on the basis of previous performance. On the other hand we have formal and structural methods, where confidence is given depending on the structural properties of the system. Structural methods exploit the knowledge about the structure of the system (the rules and fuzzy sets used in their definition) rather than the knowledge about the problem the system solves.

There is a previous extensive work on validation and verification of rule-based knowledge systems that could help us. Unfortunately, it is mainly focused on classical boolean or multivalued logic, few methods are oriented towards the evaluation of fuzzy knowledge sources based on production rules [40, 52]. The present work (based on a previous work done by the author that can be found in [23]) focuses on structural criteria and methods for fuzzy rule bases quality and reliability measurement.

We define three new "a priori" criteria to measure the quality of fuzzy systems, they are Completeness, Redundancy and consistency. They are



**Figure 2.4:** The outer square encloses  $2^X$ , the set of all the fuzzy subsets of  $X = \{x_1, x_2\}$ . We represent the fuzzy set  $F$  defined by  $m_F(x_1) = 0.5$ ,  $m_F(x_2) = 0.66$  as a point inside the unit square. The darker region contains all the proper fuzzy subsets of  $F$ , that is, all the fuzzy subsets  $F'$  such that  $F' \subseteq F$ .

based on similar criteria used in classical logic, but we generalize them to the fuzzy domain. This is mainly done making use of Kosko's set-as-points framework [41] and his subsethood theorem. The section is structured as follows: in first place we introduce the sets-as-points framework and use it to define a new fuzzy subsethood relationship. Secondly we introduce the three new structural criteria.

#### 2.4.1 The Sets-as-Points Framework.

Given a set  $X = \{x_1..x_n\}$ , we can establish a bijection between the fuzzy power set of  $X$  and the  $[0, 1]^n$  hypercube. So, each fuzzy subset  $F$  of  $X$  is represented by a single point  $p = (m_F(x_1), m_F(x_2) \dots m_F(x_n))$  in  $[0, 1]^n$ . The vertices of the hypercube will correspond to classical or crisp sets while other points will correspond to fuzzy ones (see Figure 2.4). This way of defining fuzzy sets is called the Set-as-Points framework [41], opposite to the classical [61] Sets-as-Functions definition.

The Set-as-Points framework allows us to assign a geometrical meaning to the concepts of fuzzy sets theory and to define new ones. Particularly, it is quite straightforward to define the concept of distance between two fuzzy sets as the distance between the points representing them.

**Definition 2.1** Let  $X = \{x_1..x_n\}$  be an universal set. Let  $F$  and  $G$  be two

fuzzy subsets of  $X$ . We define the family of distances  $d^p$  between  $F$  and  $G$  as:

$$d^p(F, G) = \left( \sum_{i=1}^n |m_F(x_i) - m_G(x_i)|^p \right)^{\frac{1}{p}} \quad (2.1)$$

The simplest distance is  $d^1$ , or fuzzy Hamming distance, while  $d^2$  corresponds to the fuzzy euclidean distance.

The cardinality or measure  $M$  of a fuzzy set  $F$  has a natural geometric interpretation in the Set-as-Points framework.  $M(F)$  equals the magnitude (making use of Hamming distance) of the vector drawn from the origin to the point representing the fuzzy subset  $F$ . That is,  $M(F)$  equals the fuzzy Hamming distance from the point representing  $\emptyset_X$  (the origin) to the point representing  $F$ :

$$M(F) = \sum_{i=1}^n m_F(x_i) = d^1(F, \emptyset_X) \quad (2.2)$$

#### 2.4.2 The Subsethood Relationship

The classical fuzzy containment relationship (see Appendix A) is a crisp one. That is, given an universal set  $X$  and two different fuzzy subsets of  $X$ ,  $F$  and  $G$ , at most one of  $F \subseteq G$  and  $G \subseteq F$  is true. It seems possible and desirable to introduce a fuzzy version of the containment relationship that help us measuring the degree that  $F$  and  $G$  contain each other. We will call it the **subsethood relationship**

**Definition 2.2** Let  $X = \{x_1..x_n\}$  be an universal set. Let  $F$  and  $G$  be two fuzzy subsets of  $X$ . We define the subsethood relationship between  $F$  and  $G$ , (that is, the degree that  $F$  is a subset of  $G$ ) as:

$$S(F, G) = 1 - \frac{\sum_{i=1}^n \max(0, m_F(x_i) - m_G(x_i))}{M(F)} \quad (2.3)$$

or, equivalently (Subsethood theorem):

$$S(F, G) = \frac{M(F \cap G)}{M(F)} \quad (2.4)$$

Shortly,  $S(F, G)$  measures the distance between  $F$  and the closest-to- $F$  proper subset of  $G$ . The interested reader can find several justifications of this relationship, as well as the proof of the Subsethood theorem, in [41]

### 2.4.3 Three structural criteria for fuzzy systems reliability assessment

In this section we will try to define and formalize three criteria for structural evaluation of fuzzy systems quality: completeness, redundancy and consistency. We borrow them from classical crisp logic and then we "fuzzyfy" them making use of the subethood theorem. We define each criteria in two complementary ways: locally and globally. The local value refers to a single point in the input space, while the global value refers to the overall fuzzy system. So, we will talk, for example, about the overall completeness of a fuzzy system and, also, about the consistency of a fuzzy system for a given value of the input variables.

Although several other criteria could have been defined in similar ways, we have found completeness, redundancy and consistency to be the more simple and adequate. We will return to this point in the conclusions part.

#### 2.4.3.1 Notation

Let  $U_X = \{x_1 \dots x_n\}$  and let  $U_Y = \{y_1 \dots y_m\}$  be two universal sets and  $V = \{v_1 \dots v_p\}$  and  $W = \{w_1 \dots w_q\}$  be two families of fuzzy subsets of  $U_X$  and  $U_Y$  respectively, defined by the two families of functions:

$$\begin{aligned} m_{v_i} : U_X &\rightarrow [0 \dots 1] & 1 \leq i \leq p \\ m_{w_i} : U_Y &\rightarrow [0 \dots 1] & 1 \leq i \leq q \end{aligned}$$

Let  $A$  and  $B$  be two fuzzy variables (being  $A$  the *input fuzzy variable* and  $B$  the *output fuzzy variable*) taking values on the fuzzy subsets of  $U_X$  and  $U_Y$  respectively.

Let  $K$  be a fuzzy system formed by a set  $R = \{r_1 \dots r_s\}$  of rules of the form<sup>3</sup>:

---

<sup>3</sup>We are not losing any generality by considering only rules with a single antecedent. Any rule in the form:

$$\mathbf{IF} A_1 = v_i \mathbf{AND} A_2 = v_2 \mathbf{AND} \dots \mathbf{AND} A_n = v_n \mathbf{THEN} B = w$$

is equivalent to the single antecedent fuzzy rule:

$$\mathbf{IF} (A_1, A_2, \dots, A_n) = v^* \mathbf{THEN} B = w$$

where  $v^* = v_1 \times v_2 \times \dots \times v_n$  is defined by the membership function:

$$m_{V^*}((x_1, x_2, \dots, x_n)) = \min_i (m_{v_i}(x_i))$$

$$\mathbf{IF} A = v_i \mathbf{THEN} B = w_j \quad (\text{for some } v_i \in V \text{ and } w_j \in W)$$

For each rule  $r \in R$ , we will call the *antecedent fuzzy set* of  $r$  ( $ant(r)$ ) to the fuzzy set in the **IF** part of the rule (the fuzzy set  $v_i$  in the rule above). Similarly, we will call the *consequent fuzzy set* of  $r$  ( $con(r)$ ) to the fuzzy set in the **THEN** part of the rule.

### 2.4.3.2 Completeness

Intuitively, we can see the completeness of a fuzzy system as the degree to which every possible value of the input variable is contemplated in the antecedent part of the rule set  $R$ . That is, a fuzzy system is more incomplete as there are more values of the input variable that does not fire any rule or that fire few of them weakly.

For example, suppose we have two fuzzy systems  $F$  and  $S$ .  $F$ 's rule base is formed by nine rules and  $S$ 's rule base consists of just three rules. Both  $F$ 's and  $S$ 's rules have the form:

$$\mathbf{IF} A = v_1 \mathbf{THEN} B = w_1$$

Suppose the fuzzy sets corresponding to the antecedent part of the rules are as depicted in Figure 2.5. It is quite clear that the lower definition of the antecedent sets leads to a quite incomplete fuzzy system where most values of the input variable do not fire any rule or fire a single rule weakly. On the other hand, the upper definition of the antecedent sets leads to a pretty complete system where almost every value of the input variable fires several rules almost completely.

Let  $\emptyset_{U_X}$  be the empty fuzzy subset of  $U_X$ . We will define the global completeness of the fuzzy system  $K$  as the degree to which the union of the antecedent fuzzy sets of the rule set  $R$  differs from  $\emptyset_{U_X}$ . This is the same as comparing the superposition of all the antecedent fuzzy subsets of  $R$  with  $\emptyset_{U_X}$ . Let us formalize this concept a little more.

**Definition 2.3** We define the union of the antecedent fuzzy sets of the rule set  $R = \{r_1 \dots r_s\}$  as the fuzzy subset  $ANT_R = \bigcup_{r \in R} ant(r)$  given by the function:

$$m_{ANT_R}(x_i) = \max_{r \in R} m_{ant(r)}(x_i) \quad \forall x \in U_X. \quad (2.5)$$

How can we compare  $ANT_R$  and  $\emptyset_{U_X}$ ? We can use the concept of distance between fuzzy sets as defined in Equation 2.1. The completeness of  $R$  will be proportional to the distance between  $ANT_R$  and  $\emptyset_{U_X}$ .



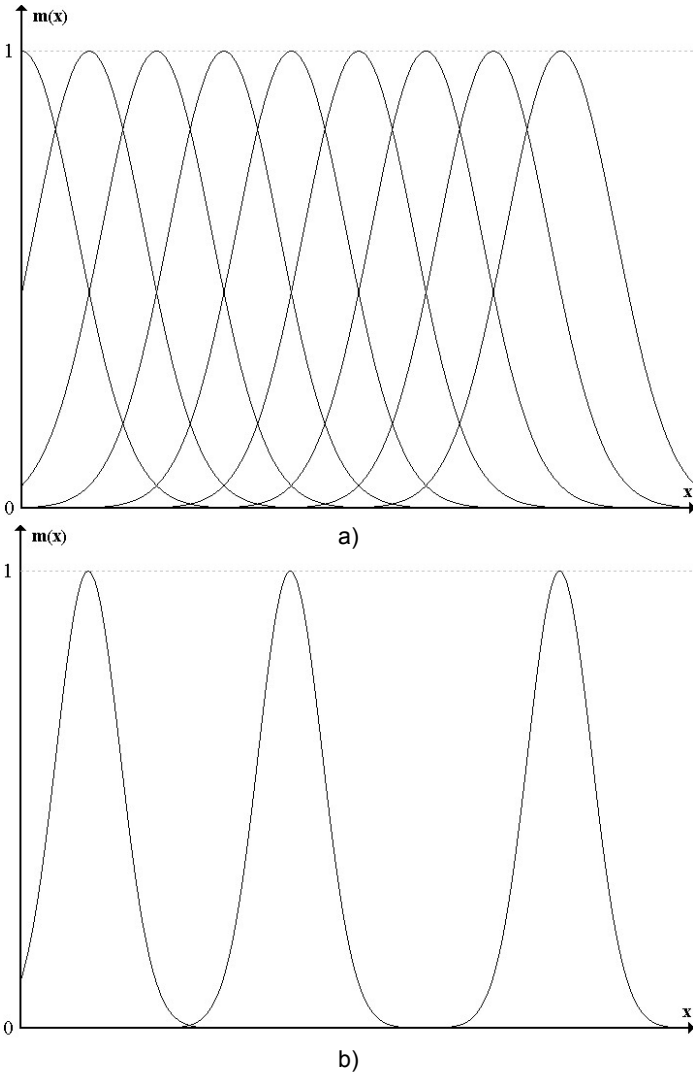


Figure 2.5: a) A quite complete antecedent set. b) A quite incomplete one.

**Definition 2.4** We define the **global completeness** of the fuzzy system  $K$ , with rule set  $R$  as:

$$COMP(K) = \frac{d(ANT_R, \emptyset_{U_X})}{d(\emptyset_{U_X}, C_{U_X})} \quad (2.6)$$

where  $COMP(K)$  is normalized by dividing  $d(ANT_R, \emptyset_{U_X})$  by the maximal distance between fuzzy subsets of  $U_X$ ,  $d(\emptyset_{U_X}, C_{U_X})$ , and  $d$  is some general distance. Particularly, for the rectangular distance, we can develop the formula to obtain:

$$COMP(K) = \frac{\sum_{x_i \in U_X} m_{ANT_R}(x_i)}{|U_X|} = \frac{\sum_{i=1}^n m_{ANT_R}(x_i)}{n} \quad (2.7)$$

To restrict the definition to a single point in the input space is quite straightforward:

**Definition 2.5** We define the **completeness** of the fuzzy system  $K$ , with rule set  $R$ , at the point  $x_i \in U_X$  of the input space as the value of the membership function of  $ANT_R$  in this point:

$$COMP(K, x_i) = m_{ANT_R}(x_i) \quad \forall x_i \in U_X. \quad (2.8)$$

We can observe that, when using rectangular distances, the global completeness of a fuzzy system is the average of the completeness of the system at each point of the input space. This does not hold, however, in the general case.

### 2.4.3.3 Redundancy

Redundancy in a rule base refers to the existence of clusters of very similar rules. If the rule base has been built from a set of examples, redundancy is a favorable feature, because it indicates a degree of regularity in the system (similar inputs result in similar outputs). On the other hand, depending on the propagation model used by the fuzzy system, a big number of redundant rules just burdens the output computation process without substantially affect the overall response of the system.

The concept of rule redundancy relies on the concept of rule subsumption. Let us introduce it.

**Definition 2.6** Given two rules  $r_1$  and  $r_2$  of the form:

$$\begin{aligned} r_1 &: \text{IF } A = v_1 \text{ THEN } B = w_1 \\ r_2 &: \text{IF } A = v_2 \text{ THEN } B = w_2 \end{aligned}$$

we will say that  $r_2$  is **subsumed** by  $r_1$  if and only if:

- $v_2 \subseteq v_1$ . The antecedent part of rule  $r_2$  is more constricting than the antecedent part of rule  $r_1$ 's.
- $w_1 \subseteq w_2$ . The consequent part of rule  $r_1$  is more specific than the consequent part of rule  $r_2$ .

where  $\subseteq$  is the classical fuzzy inclusion relationship (that is,  $a \subseteq b$  if and only if  $m_a(x) \leq m_b(x)$  for all  $x \in U_X$ ). From this definition, subsumption is clearly a crisp concept. Given two rules  $r_1$  and  $r_2$ , one and only one of the statements “ $r_1$  is subsumed by  $r_2$ ” and “ $r_1$  is not subsumed by  $r_2$ ” is true. However, from Figure 2.6, where we can see three examples of rule pairs with different values for  $v_1, v_2, w_1$  and  $w_2$ , it seems clear that subsumption is a matter of degree. We need, then, to extend the concept of subsumption to the fuzzy domain. We can derive a fuzzy subsumption relationship by using the fuzzy subsethood relationship in the following way:

$$sub(r_1, r_2) = \min(S(v_2, v_1), S(w_1, w_2)) \quad (2.9)$$

Clearly, given two rules, the more one of them subsumes the other, the more redundant they are. So, now we can define the pairwise redundancy of two rules  $r_1$  and  $r_2$  as :

$$pw\_red(r_1, r_2) = \max(sub(r_1, r_2), sub(r_2, r_1)) \quad (2.10)$$

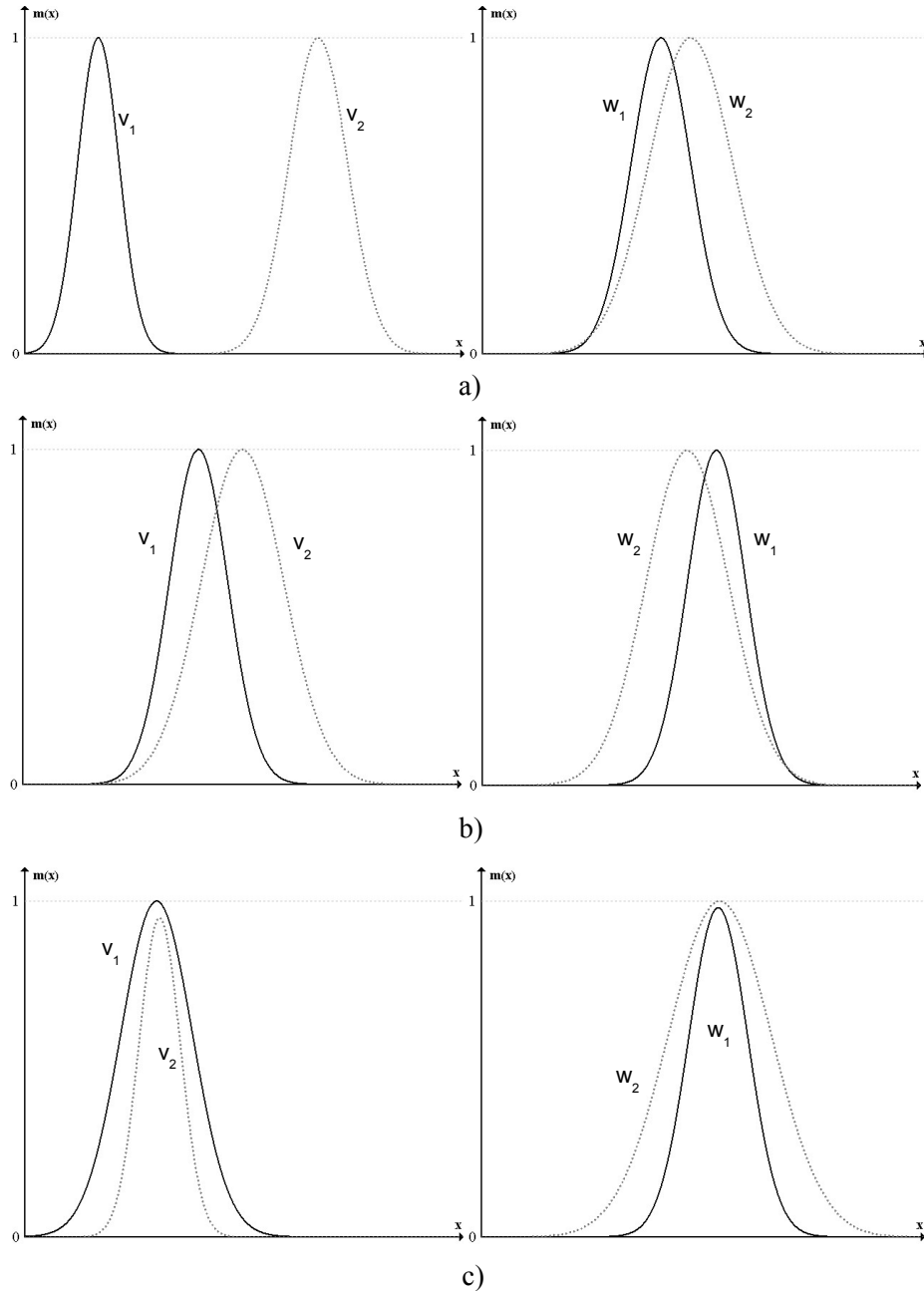
Furthermore it is clear that, in order to affect the output of the system, redundant rules have to fire. So, if we want to compute the redundancy of a pair of rules for a given value of the input variable we have to weight the redundancy found using Equation 2.10 by a value proportional to the degree to which both rules are firing, that is, the degree to which the value of the input variable belongs to the antecedent fuzzy sets of the two rules. Thus, given a point  $x_i$  in the input space, we can define the pairwise redundancy of two rules  $r_1$  and  $r_2$  at the point  $x_i$  as:

$$pw\_red(r_1, r_2, x_i) = \min(m_{ant(r_1)}(x_i), m_{ant(r_2)}(x_i)) \cdot pw\_red(r_1, r_2) \quad (2.11)$$

We can now define the global and local redundancy of a fuzzy system as the average value of the redundancies for each unordered pair of different rules:

**Definition 2.7** We define the **global redundancy** of a fuzzy system  $K$  with rule set  $R = \{r_1 \dots r_s\}$  as:

$$RED(K) = \binom{s}{2}^{-1} \cdot \sum_{1 \leq i, j \leq s, i < j} pw\_red(r_i, r_j) \quad (2.12)$$



**Figure 2.6:** a) Rule 1 and rule 2 are not redundant at all. b) Rule 2 is partially subsumed by rule 1. c) Rule 2 is completely subsumed by rule 1.

**Definition 2.8** Conversely, we define the **redundancy** of a fuzzy system  $K$  with rule set  $R = \{r_1 \dots r_s\}$  at a point  $x_i$  of the input space  $U_X$  as:

$$RED(K, x_i) = \binom{s}{2}^{-1} \cdot \sum_{1 \leq i, j \leq s, i < j} pw\_red(r_i, r_j, x_i) \quad (2.13)$$

#### 2.4.3.4 Consistency

Inconsistency in a rule base refers to the existence of rules with similar antecedent parts and different consequent parts. Inconsistent rules are not favorable features for a rule base and can indicate a irregular, aleatory or chaotic behavior in the modeled system, as well as the occurrence of mistakes in the modeling process.

In classical logic, we say that two rules are inconsistent if they have the same antecedents and contradictory consequents; we can extend the concept to the fuzzy domain in the following way:

**Definition 2.9** Given two fuzzy rules  $r_1$  and  $r_2$ :

$$\begin{aligned} r_1 &: IF A = v_1 THEN B = w_1 \\ r_2 &: IF A = v_2 THEN B = w_2 \end{aligned}$$

we will say that  $r_2$  is **inconsistent** with  $r_1$  if and only if :

- $v_2 \subseteq v_1$ . The antecedent part of rule  $r_2$  is included in the antecedent part of rule  $r_1$ .
- $w_2 \subseteq w_1^c$ . The consequent part of rule  $r_2$  is included in the complement of the consequent part of rule  $r_1$ .

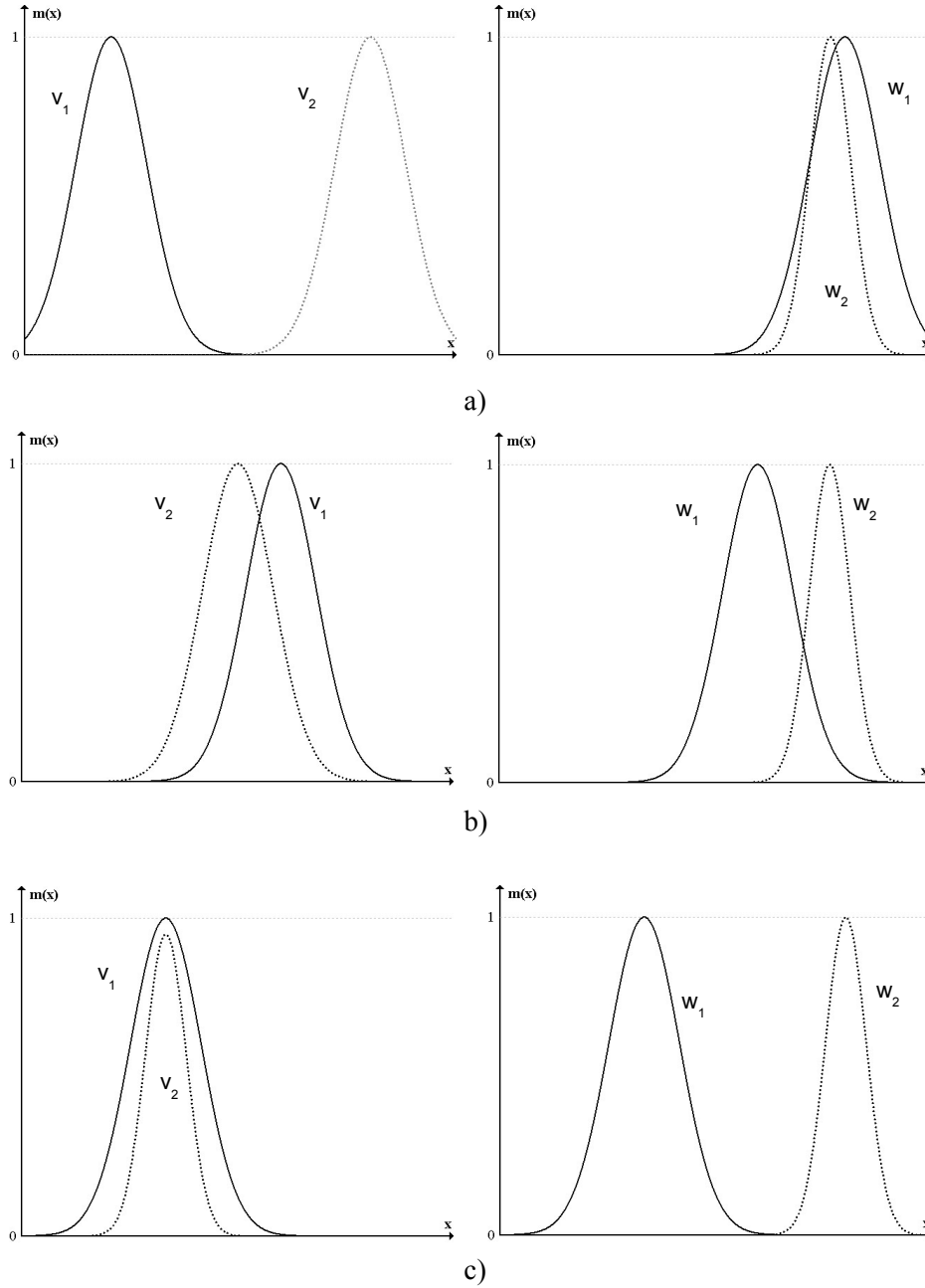
where  $a \subseteq b$  is the classical fuzzy inclusion relationship (that is  $a \subseteq b$  if and only if  $m_a(x) \leq m_b(x)$  for all  $x \in U_X$ ).

Inconsistency as defined in Definition 2.9 is a crisp concept. However, in Figure 2.7 we can see three different examples of rule pairs which make quite apparent the fuzzy character of inconsistency. As in the case of redundancy, we need to extend the concept of consistency to the fuzzy domain.

Like subsumption, inconsistency is crisp because of the fuzzy inclusion relationship. Again, we can derive a fuzzy inconsistency relationship for a pair of rules  $r_1$  and  $r_2$  using the fuzzy subsethood relationship as follows:

$$inc(r_1, r_2) = \min(S(v_2, v_1), S(w_2, w_1^c)) \quad (2.14)$$

And now we can define the pairwise inconsistency of two rules  $r_1$  and  $r_2$  as the maximum of the mutual inconsistencies:



**Figure 2.7:** a) A consistent pair of rules. b) Rule 1 and rule 2 are partially inconsistent with each other. c) Rule 2 is completely inconsistent with rule 1.

$$pw\_inc(r_1, r_2) = \max(inc(r_1, r_2), inc(r_2, r_1)) \quad (2.15)$$

As it was the case with redundancy, if we want to compute the pairwise inconsistency of two rules at a given point of the input space, it is necessary to weight the value found using Equation 2.15 proportionally to the degree to which both rules fire at that point. So, given a point  $x_i$  in the input space, we define the pairwise inconsistency of two rules  $r_1$  and  $r_2$  at the point  $x_i$  as:

$$pw\_inc(r_1, r_2, x_i) = \min(m_{ant(r_1)}(x_i), m_{ant(r_2)}(x_i)) \cdot pw\_inc(r_1, r_2) \quad (2.16)$$

We are interested in computing consistencies, rather than inconsistencies. Values obtained from Equations 2.15 and 2.16 rank between 0 and 1. This allows us to straightforwardly define the pairwise consistency of two rules  $r_1$  and  $r_2$  as:

$$pw\_con(r_1, r_2) = 1 - pw\_inc(r_1, r_2) \quad (2.17)$$

and the pairwise consistency of two rules  $r_1$  and  $r_2$  at a given point  $x_i$  as:

$$pw\_con(r_1, r_2, x_i) = 1 - pw\_inc(r_1, r_2, x_i) \quad (2.18)$$

Finally, we can define the global and local consistency of a fuzzy system as the average value of the consistencies for each unordered pair of different rules:

**Definition 2.10** We define the **global consistency** of a fuzzy system  $K$  with rule set  $R = \{r_1 \dots r_s\}$  as:

$$CON(K) = \binom{s}{2}^{-1} \cdot \sum_{1 \leq i, j \leq s, i < j} pw\_con(r_i, r_j) \quad (2.19)$$

**Definition 2.11** Conversely, we define the **consistency** of a fuzzy system  $K$  with rule set  $R = \{r_1 \dots r_s\}$  at a point  $x_i$  of the input space  $U_X$  as:

$$CON(K, x_i) = \binom{s}{2}^{-1} \cdot \sum_{1 \leq i, j \leq s, i < j} pw\_con(r_i, r_j, x_i) \quad (2.20)$$

#### 2.4.4 The reliability function

We have defined a total of six different criteria for fuzzy rule base quality assessment, three refer to the quality of the overall rule base and the other three refer to the quality of the rule base for a given a point in the input space. The combination of these six criteria into a single reliability function can be done reasonably in a big number of ways. In any case it is no evidence of such a combination to be more useful than using the criteria separately. We will come back to this point in our case study.

## 2.5 Case Example: Calibration of an Ammeter

An ammeter is a measuring instrument used to measure the electric current in a circuit. Electric currents are measured in amperes, hence the name. Let's use an idealized ammeter as a case study in order to exemplify the concepts introduced in the previous sections as well as to gain some insight into their applicability and appropriateness.

Suppose we have a cheap, inexact ammeter  $A$  that gives wrong readings in a way that depends on the intensity of the current and some other external variable (temperature, let's say). Suppose the error in the ammeter measurement to be an aleatory variable with mean and variance directly proportional to current intensity and room temperature. Suppose, finally, that we have an exact ammeter that allows us to determine the error committed by the inexact ammeter  $A$  in a set of measurements. That is, we have a set  $E = \{e_1, e_2, \dots, e_n\}$  of examples in the form of triplets:

$$e_i = (I_i, T_i, R_i)$$

Where  $I_i$  stands for the intensity measured by  $A$  at instant  $i$ ,  $T_i$  is the temperature at this instant and  $R_i$  is the true current intensity as measured by the unbiased ammeter at the same instant  $i$ . Can we use this knowledge to build a fuzzy contextual filter able to correct  $A$ 's readings in subsequent measurements in such a way that the expected value of the error diminishes? Moreover, how to compute the reliability criteria introduced in this chapter and how to use them to assess the reliability of the corrections introduced by the fuzzy contextual filter? We will try to give an answer to these questions in the next sections.

### 2.5.1 Modeling the ammeter

In order to collect the set of examples  $E$  and evaluate the performance of the fuzzy contextual filter, we need a model of the behavior of the ammeter  $A$ , that is, a function relating temperatures ( $T$ ) and real current intensities ( $R$ ) to current intensities ( $I$ ) as measured by  $A$ . A reasonable model to choose is the following one:

$$I(T, R, \alpha_1, \alpha_2, K) = R + \alpha_1 + N\left(\alpha_2 \cdot R, (K \cdot T)^2\right) \quad (2.21)$$

where  $T$  and  $R$  are the actual temperature and current intensity, respectively and  $\alpha_1, \alpha_2$  and  $K$  are parameters that allow the modulation of the model's behavior. Parameter  $\alpha_1$  represents a constant bias in the ammeter and  $N\left(\alpha_2 \cdot R, (K \cdot T)^2\right)$  is a normal random variable with mean  $\alpha_2 \cdot R$  and standard deviation  $K \cdot T$ . Parameter  $K$ , therefore, establishes how "noisy" is the error committed by  $A$ . For  $K = 0$ , the error will be a constant  $\alpha_1 + \alpha_2 \cdot R$



and for greater values of  $K$  the error will fluctuate randomly with amplitude proportional to the product of the magnitude of the parameter and the temperature.

In order to avoid having to manage too many combinations of parameter values through all the set of simulations,  $\alpha_1$  and  $\alpha_2$  have been given adequate values, namely 4 and 0.2, respectively. On the other hand, the rank in which the input variables temperature and current intensity can take values has been limited to  $[0, \dots, 50]$  in some well suited units.

Figure 2.8 shows representative examples of the error committed by ammeter  $A$  for different values of the parameter  $K$ , namely 0, 0.01, 0.05, 0.1, 0.2 and 0.5. As can be seen, for  $K = 0$  the error is independent from the temperature. As  $K$ 's value increases, so does the influence of the temperature in the overall error. At high temperatures and for large values of the parameter  $K$ , measurements of the ammeter are essentially random.

### 2.5.2 Building the Fuzzy Contextual Filter

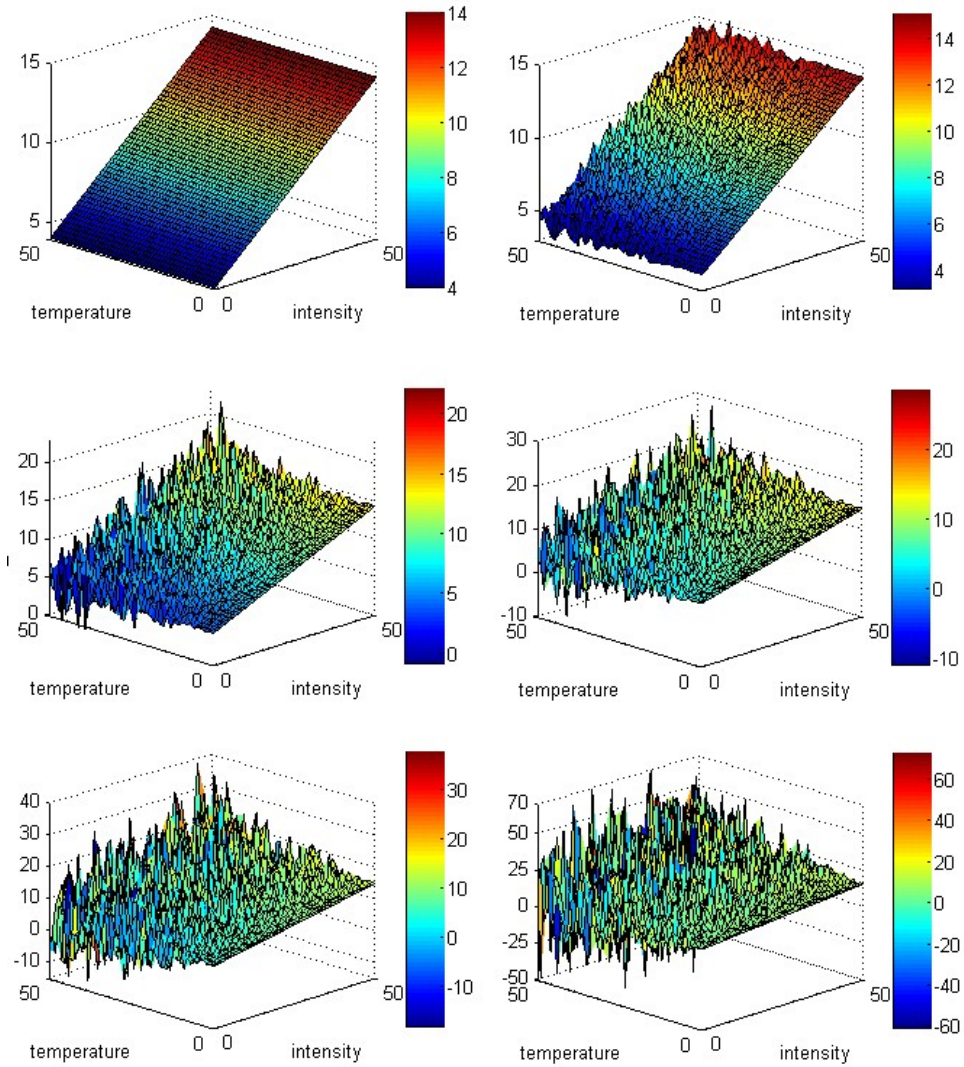
We can see the problem represented in Figure 2.9. We have an ammeter that gives wrong readings, the error depending, in some way unknown to us<sup>4</sup>, on the magnitude of the current intensity and the ambiance temperature. The problem consists in building a FCF which takes  $A$ 's readings as the input variable and the temperatures as the only context variable and produces, as output variable, an estimation of the correct value of the current intensity, jointly with some kind of measure of the accuracy of the estimation.

Building the corrective module of the FCF from the set of examples  $E$  is a quite straightforward task. Recall that each example  $e_i$  has the form  $e_i = (I_i, T_i, R_i)$ , where  $I_i$  stands for the intensity measured by  $A$  at instant  $i$ ,  $T_i$  is the temperature at this instant and  $R_i$  is the true current intensity as measured by the unbiased ammeter at the same instant  $i$ . We can produce a fuzzy rule from each one of the examples interpreting the values  $I_i$ ,  $T_i$ , and  $R_i$  as fuzzy numbers much in the same way as stated in section 2.3. So, the set  $E$  of examples will define the dynamic fuzzy rule base of the corrective module of the FCF. The initial rule base is not needed in this example so it will be considered empty.

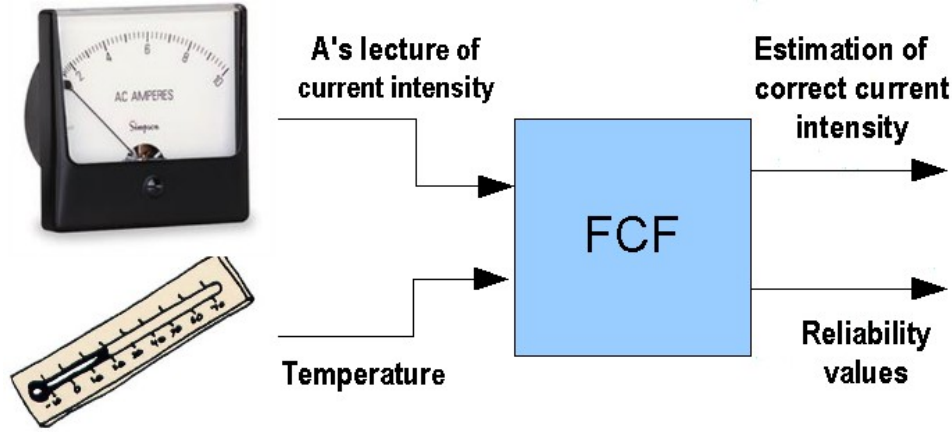
Let us elaborate a bit more upon the process of construction of the dynamic rule base. For our purpose, given a real number  $a$ , we will define  $F_a$ , the fuzzy number associated to  $a$ , as the fuzzy set over the reals defined by the following membership function:

---

<sup>4</sup>We use the model defined in the previous section to collect the set of examples and to simulate the behavior of the ammeter but, obviously, we can't use it to build the FCF



**Figure 2.8:** Error committed by ammeter  $A$  for different values of the parameter  $K$ . From top to bottom and from left to right, the value of  $K$  is 0, 0.01, 0.05, 0.1, 0.2 and 0.5



**Figure 2.9:** Representation of the problem

$$m_{F_a}(x) = \begin{cases} e^{-\frac{(x-a)^2}{c^2}} & \text{if } |x - a| < 3c, \\ 0 & \text{otherwise.} \end{cases} \quad (2.22)$$

a sort of Gaussian function with its tails truncated (that is, the value of the function is set to zero at every point far away enough from the center of the bell) in which the parameter  $c$  regulates the amplitude of the bell, as well as the truncation point<sup>5</sup>.

Now, for each example  $e_i = (I_i, T_i, R_i)$ , the following rule is added to the dynamic rule base:

If  $A$ 's reading is  $F_{I_i}$  and temperature is  $F_{T_i}$  then *correctIntensity* is  $F_{R_i}$

This process produces a fuzzy rule base that can be used, after the addition of the appropriate fuzzification and defuzzification mechanisms, as a fuzzy system able to estimate correct current intensity values from  $A$ 's current intensity readings and environment temperatures. In this example, the whole process of fuzzification, propagation and defuzzification can be summarized as follows: given a current intensity  $x$  measured by the ammeter  $A$  and a value  $y$  of the environment temperature, the estimation  $r(x, y)$  of the correct current intensity (the output of the fuzzy system) is computed using the following formula:

---

<sup>5</sup>The reason for this truncation is mainly practical, if it did not exist, every rule of the rule base would fire, even if to a minute degree, for every pair of input variables. On the other hand, the truncation will make much more efficient the computation of the reliability criteria introduced in Section 2.4.

$$r(x, y) = \frac{\sum_{(I,T,R) \in E} \min(m_{F_I}(x), m_{F_T}(y)) \cdot R}{\sum_{(I,T,R) \in E} \min(m_{F_I}(x), m_{F_T}(y))} \quad (2.23)$$

Figure 2.10 shows six examples of the error committed by the fuzzy system in the particular case of parameter  $K$  in equation 2.21 taking value 0 (comparisons for other values of the parameter  $K$  will be seen later in the Results section). It is interesting to compare those errors with the top left graph of Figure 2.8, which shows the error committed by the ammeter  $A$  for the same value of the parameter  $K$ .

Figure 2.10 shows error graphs for different values of the cardinal of the examples set and the parameter  $c$  (from Equation 2.22, the parameter that regulates the fuzzy number width). From left to right and top to bottom, the three first graphs correspond to a set  $E$  of 100 examples and the three last ones to a set  $E$  of 1000 examples. For each of the groups of three, the values of the parameter  $c$  are 1, 2 and 4 respectively. The importance of both parameters is quite clear: in the cases of fuzzy systems with few rules or small fuzzy number width there is a lot of input pairs that do not fire any rule, thus giving output 0 and producing a big error<sup>6</sup>. On the other hand, when both values are high enough to make every pair of input values fire one or more rules (100 examples and  $c = 4$ , 1000 examples and  $c = 2$  and 1000 examples and  $c = 4$ ), the error committed by the fuzzy system is much lower than the error committed by ammeter  $A$ .

### 2.5.3 The reliability calculation module

In order to test and compare their usefulness, all three reliability criteria introduced in Section 2.4, (completeness, redundancy and consistency), have been implemented in the simulation.

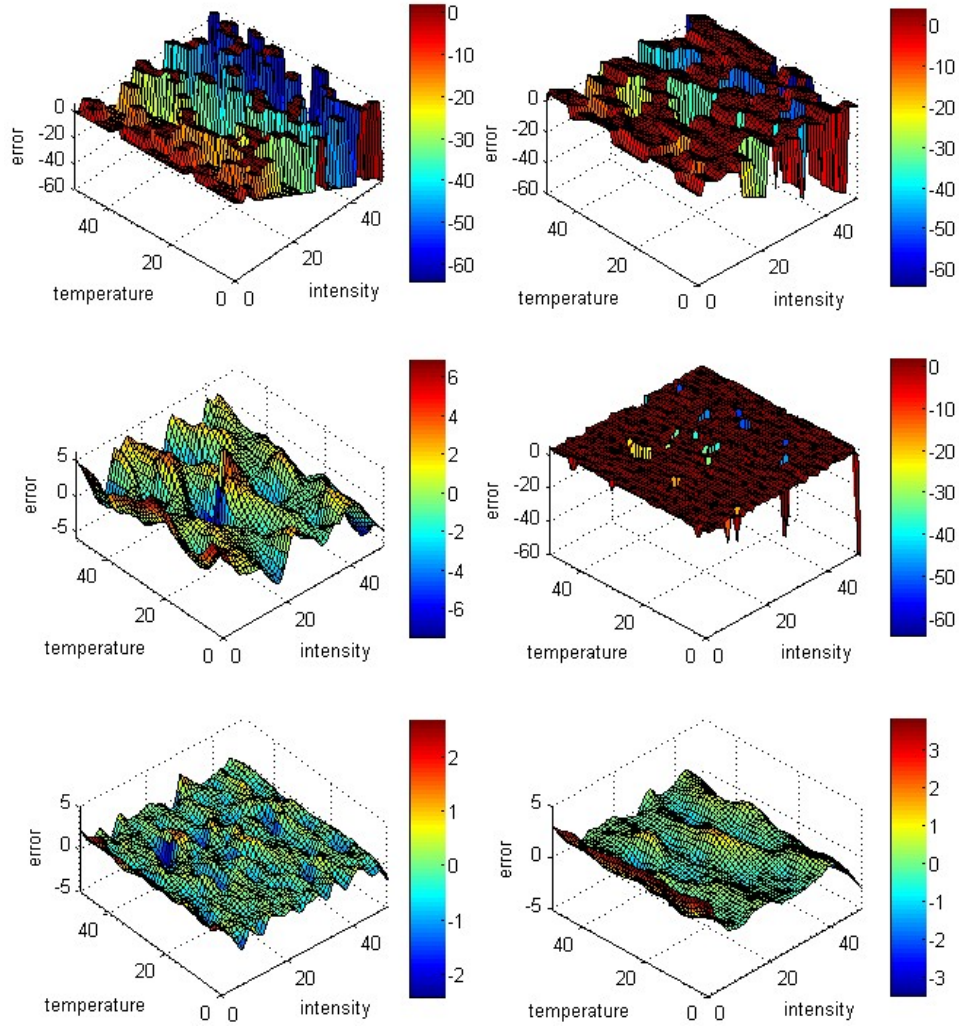
### 2.5.4 Results

A total of 72 simulations have been carried out for different values of the parameters  $n$  (number of examples and rules),  $c$  (width of the fuzzy numbers) and  $K$  (standard deviation of the random part of the error, see Equation 2.21). Specifically, the values tested are  $n = 100, 500, 1000$ ,  $c = 1, 2, 4$  and  $K = 0, 0.01, 0.025, 0.05, 0.1, 0.2, 0.5, 1$ .

For each combination of parameter values, each simulation consisted in the following steps:

---

<sup>6</sup>This issue could be easily solved by, for example, establishing  $R_i = I_i$  when none of the fuzzy rules in the rule base fires. However this is not necessary to show our point



**Figure 2.10:** Graphs of the errors committed by the FCF for  $K = 0$  and for different values of the parameters  $c$  and number of rules. From left to right and top to bottom, graphs 1,2 and 3, 100 examples, graphs 4, 5, and 6, 1000 examples. Graphs 1 and 4,  $c = 1$ , graphs 2 and 5,  $c = 2$  and graphs 3 and 6,  $c = 4$ .

1. Populate the example set  $E$  with  $n$  examples according to Equation 2.21 for random values of  $I$  and  $T$  in the real interval  $[0, \dots, 50]$ .
2. Build the fuzzy rule base as explained in section 2.5.2.
3. Generate a new example set  $S$  (we will call it the reference example set) using Equation 2.21 to simulate the behavior of ammeter  $A$  for every pair  $(I, T) \in \{0, 1, \dots, 50\} \times \{0, 1, \dots, 50\}$ .
4. Compute the error committed by ammeter  $A$  for each example  $(I, T, R)$  in the reference example set  $S$  as:

$$err_A(I, T, R) = |I - R| \quad (2.24)$$

5. Compute the output of the FCF for all pairs  $(I, T) \in \{0, 1, \dots, 50\} \times \{0, 1, \dots, 50\}$ .
6. Calculate the error committed by the FCF for each example  $(I, T, R)$  in the reference example set  $S$  as:

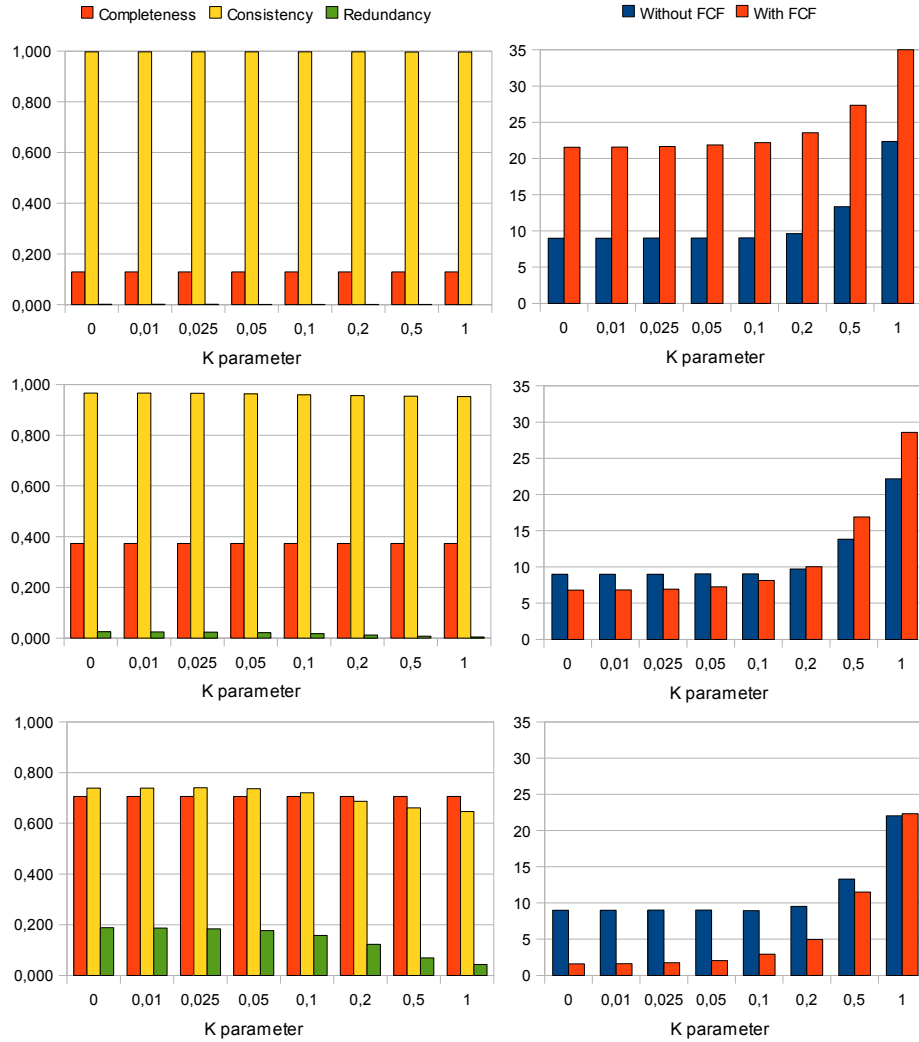
$$err_{FCF}(I, T, R) = |r(I, T) - R| \quad (2.25)$$

where  $r(I, T)$  is the output of the FCF for the input pair  $(I, T)$  as shown in Equation 2.23.

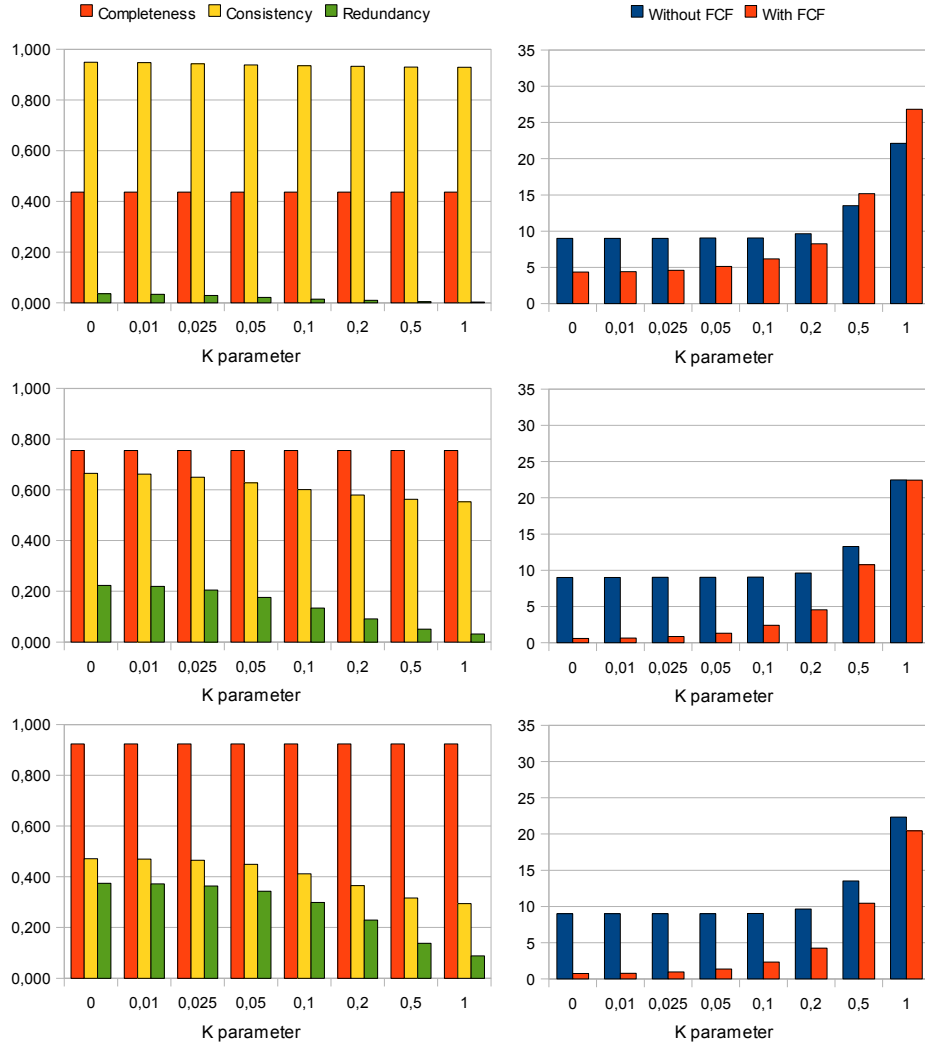
7. For the fuzzy rule base built in point 2, compute all the reliability criteria for FCFs introduced in section 2.4.3.
8. Compare the errors committed by the FCF and the ammeter  $A$  and relate them to the reliability criteria calculated in the previous step.

Over the next pages, we can see the results of the simulations. Figures 2.11, 2.12 and 2.13, relate the errors committed by ammeter  $A$  and the FCF using the global reliability criteria for different values of the parameters  $n$  (number of examples in the example set  $E$  or rules in the rule base),  $c$  (width of the fuzzy numbers) and  $K$  (standard deviation of the random part of the error, see Equation 2.21).

Figure 2.11 corresponds to a 100 examples rule base. Each row shows the relationship between the reliability criteria values for the rule base and the error committed by ammeter  $A$  for the reference example set  $S$ , with and without making use of the FCF. Each row correspond to a different value of the parameter  $c$ , which gets the values 1, 2 and 4 in the upper, middle and bottom rows respectively. It can be observed how both errors depend directly on the value of  $K$ . The figure shows quite dramatically the importance of parameter  $c$  in the case of a rule base with few rules: if  $c$ 's value is small, there is a lot of points in the input space which do not fire any rule in the rule

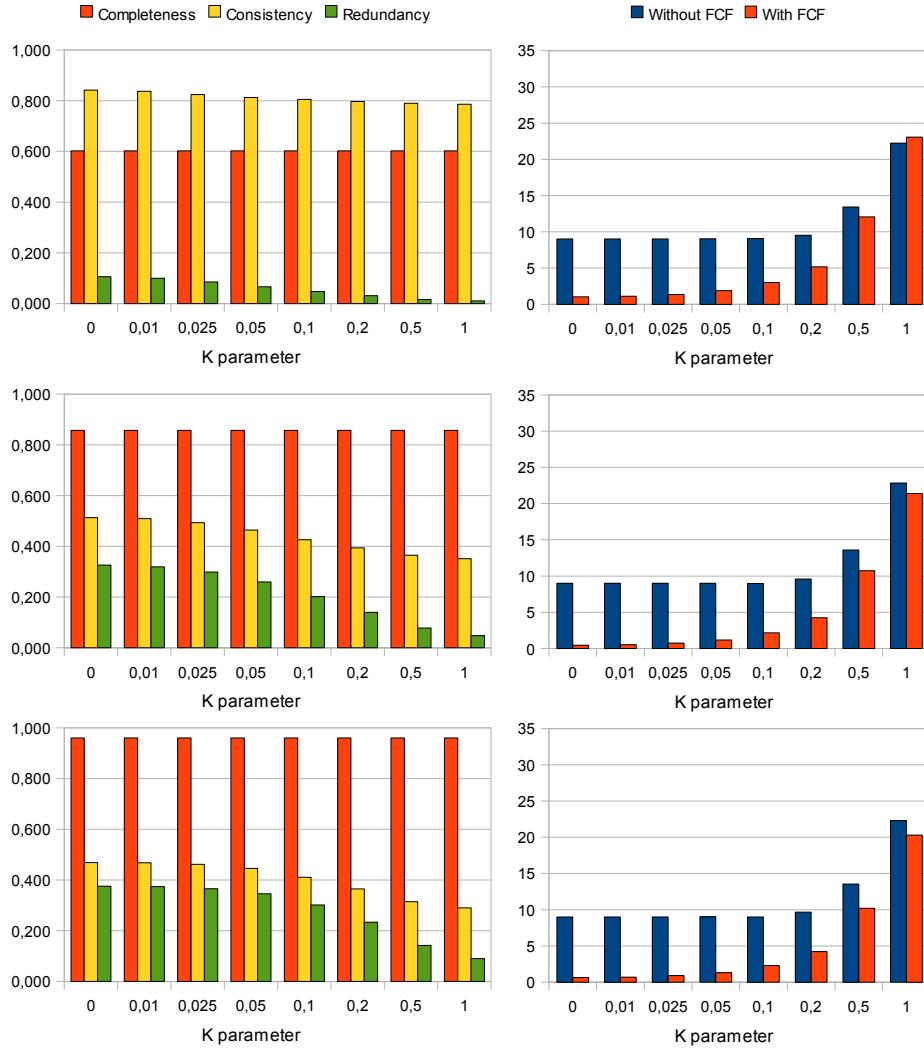


**Figure 2.11:** Global reliability criteria values vs. errors committed by ammeter A with and without FCF for a fuzzy rule base built from 100 examples and for different values of the parameters  $K$  and  $c$ . Upper row:  $c = 1$ , middle row:  $c = 2$ , bottom row:  $c = 4$ .



**Figure 2.12:** Global reliability criteria values vs. errors committed by ammeter A with and without FCF for a fuzzy rule base built from 500 examples and for different values of the parameters K and c. Upper row: c = 1, middle row: c = 2, bottom row: c = 4.





**Figure 2.13:** Global reliability criteria values vs. errors committed by ammeter  $A$  with and without FCF for a fuzzy rule base built from 1000 examples and for different values of the parameters  $K$  and  $c$ . Upper row:  $c = 1$ , middle row:  $c = 2$ , bottom row:  $c = 4$ .

base, this resulting in a big overall error. This can be seen in the upper row of the figure, corresponding to a value 1 for the parameter  $c$ . Using a FCF in this case results in a bigger error than not using it at all. All three reliability criteria can help in detecting this situation, though. Both consistency and redundancy get extreme values (near 1 and 0 respectively) due to the fact that very few points in the input space make more than a rule fire at once. On the other hand, more importantly, the completeness value of the rule base is very low.

As can be seen in the middle and bottom rows of the figure, as the width of the fuzzy numbers increases, so does the completeness of the FCF and the difference between the error committed by ammeter  $A$  when not using the FCF and the error committed when using it. Finally, for values of the parameter  $c$  high enough (bottom row,  $c = 4$ ), the performance of the ammeter when using the FCF is clearly superior to its performance without using it.

Figures 2.12 and 2.13 show information similar to Figure 2.11 when using 500 and 1000 examples in the construction of the FCF rule base, respectively. In both cases, the relatively big number of fuzzy rules makes the completeness of the fuzzy rule base increase enough to avoid the misbehavior observed in the top row of Figure 2.11. It can be observed, too, how the increase in the completeness value provoked by the increment of the parameter  $c$  translates itself in a better performance of the ammeter  $A$ , when using the FCF. It is also interesting to note how, for fixed values of the parameter  $c$  (and, thus, constant completeness value), higher values in the FCF error (directly related to the the  $K$  parameter) correspond to lower values for the consistency and redundancy criteria.

To summarize, two observations can be done with respect to the results shown in Figures 2.11, 2.12 and 2.13:

- Generally, the global performance of the ammeter is distinctively much better when using the FCF. This is specially true for small values of the parameter  $K$ . For big values of the parameter (corresponding to big variances in the random part of Equation 2.21) the improvement, if it exists at all, is not so notorious.
- It is clear that the global completeness of the fuzzy rule base is a good indicator of the overall performance of the FCF. On the other hand, when the global completeness of the rule base is high enough and the parameters  $c$  (fuzzy number width) and  $n$  (number of rules) are such that there is a reasonable probability for a random point in the input space to make two or more rules fire, it is a clear correlation between global consistency and global redundancy criteria (specially the last one) and the error committed by the FCF. The distinct merit of both

global criteria is not at all clear. Redundancy seems to have the edge due to the larger rank of values it takes over the different values of the parameter  $K$  but, on the other hand, they seem to be highly correlated.

Figures 2.14 to 2.19 show several <sup>7</sup> detailed graphs of the local consistency, redundancy and completeness reliability criteria values, jointly with graphs of the errors committed by the uncalibrated ammeter compared to the ammeter using the FCF, for different values of the parameters  $n$  (number of rules),  $c$  (fuzzy number width) and  $K$  (random error standard deviation).

Figures 2.14, 2.15 and 2.16 show the results for the cases ( $n = 100, c = 4, K = 0$ ), ( $n = 100, c = 4, K = 0.05$ ) and ( $n = 100, c = 4, K = 0.5$ ). The most interesting part of the figures is, perhaps, the completeness graph, in the middle row. It is exactly the same graph for the three figures (that is, the completeness value in each intensity-temperature point  $(I, T)$  of the input space is the same). This is due to two reasons: in one hand, the set of intensity-temperature pairs used to build the three set of examples (this is step 1 in the simulation, see Subsection 2.5.4) is the same in the three cases. Because of this, left parts of the rules in the three rule bases are equal (This is purposely this way in order to ease comparisons. Right parts will be different due to the different values of  $K$ , which will introduce errors of different expected magnitude in each case). The second reason is that, by definition (see Subsection 2.4.3.2), completeness of a rule base depends only upon the left parts (the antecedents) of the rules, which, in our case, are determined by the aforementioned intensity-temperature pairs. Completeness of a rule base mainly depends upon the two parameters  $n$  and  $c$ . A larger number of rules and a larger amplitude for the fuzzy numbers defining the antecedents of the fuzzy rules in the rule base will tend to produce a smaller probability that a random intensity-temperature input pair did not fire any rule and, thus, larger values for the global and local completeness criteria.

It is easy to recognize the left parts of the rules of the rule base in the completeness graph. Each of them corresponds to a little brown square in the graph (the width of the square depending on the parameter  $c$ ). It is also easy to see that there are big areas in the input space whose completeness value equals zero. These areas correspond to the points in the input space (that is, the intensity-temperature pairs) that do not fire any rule in the fuzzy rule base. These areas can be observed in the bottom-right graph of the figures to correspond to high error areas for the ammeter using the FCF (the center and the right side of the graph, mainly). On the other hand, as could be expected, the error committed by the ammeter using FCF tends to

---

<sup>7</sup>The complete set of results can be found in Appendix B.

be low in high completeness zones of the input space. This is particularly true in areas where the random error part of the ammeter lecture (that is, the third term of the right part of Equation 2.21) is expectedly low, and tends to become less certain in areas corresponding to expected higher random error (be it because the value of the temperature or the value of the random error standard deviation  $K$ ). Observe, for example, Figure 2.14. In it, the value of the parameter  $K$  equals zero and, consequently, from Equation (2.21), the error committed by the uncalibrated ammeter is not random, and has a value  $\alpha_1 + \alpha_2 \cdot R$ , as can be seen in the lower left graph of the figure. The graph corresponding to the error committed by the FCF (lower row to the right) shows an evident correspondence between areas of large completeness and areas of small error, independently of the temperature. This is not so true for figures 2.15 and 2.16, where the increment of  $K$ 's values (to 0.05 and 0.5), increase the influence of the random component of the error at higher temperatures, as can be seen in the bottom left graphs of the figure. This translates itself to bigger errors in the ammeter using the FCF. In particular, in Figure 2.16, where  $K = 1$ , FCF error is quite independent from completeness except for points with very low temperature.

For redundancy and consistency criteria to be meaningful at a given input point, it is necessary that two or more rules fire (the more the better, ideally). So, those two criteria will have greater usefulness and significance in rule bases with high global completeness values and in points of the input space with high local completeness. In the case of Figures 2.14, 2.15 and 2.16, due to the fact of the low completeness of the rule base (only 100 rules) it is difficult to observe a relation between redundancy, consistency and the error committed by the ammeter using the FCF. It is nevertheless possible to observe a general decrement in the consistency and redundancy values as parameter  $K$  increases. This is more patent in areas corresponding to high temperatures.

Figures 2.17, 2.18 and 2.19 show a similar series of results for the cases ( $n = 1000, c = 2, K = 0$ ), ( $n = 1000, c = 2, K = 0.05$ ) and ( $n = 1000, c = 2, K = 0.1$ ). The rule bases have 1000 rules each and, as can be expected, even if the value of the parameter  $c$  is now smaller than before, completeness values are larger at almost every point in the input space. Comments about completeness stated about the anterior set of figures also apply to this set: on one hand, areas of low completeness in the input space correspond to high error areas for the ammeter using the FCF. On the other hand, the error committed by the ammeter using the FCF tends to be low in high completeness zones of the input space, depending of the relative magnitude of the random part of the error committed by the uncalibrated ammeter.

The problem regarding the effectiveness of the completeness criteria lies, then, in areas of the input space with high completeness that correspond to

high expected error zones for the uncalibrated ammeter (high temperature zones). In them, the filtered value given by the FCF will have a big completeness value (a lot of rules will fire) and also a high expected error. consistency and redundancy criteria will help us to detect such zones in the input space.

Take a look, for example, to Figure 2.17. In it, the value of the parameter  $K$  is zero, this meaning that the random part in the error committed by the uncalibrated ammeter is a constant depending only upon the current intensity (that is, the same intensity-temperature input pair always will correspond to the same output. See Equation (2.21)). Several considerations can be done:

- As can be seen in the corresponding consistency and redundancy graphs (upper row of the figure), local consistency and redundancy values are relatively high. They are also highly correlated with completeness values, that is, they are high in areas of high completeness and low in areas of low completeness.
- No significant difference can be seen across the whole input space regarding the mean values of local consistency and redundancy values (that is, they are independent from temperature values). That could be expected, given the absence of a random component in the error.
- Comparing the two graphs in the bottom row of the figure, we can see that the ammeter using FCF performs much better than not using it. The FCF is able to correct the intensity measure output by the ammeter to give very approximately the true value across the whole input space.

Let us take a look, on the other hand, to Figures 2.18 and 2.19 . In them, the values of parameter  $K$  are 0.05 and 0.1 respectively. This translates itself in the appearance of a random component in the error made by the ammeter which depends upon the value of the temperature. The higher the temperature, the higher the random error in absolute value. This can be seen in the graphs representing the total error made by the ammeter when not using the FCF, in the lower left corner of the figures 2.18 and 2.19. The random component of the error also reflects itself in the consistency and redundancy graphs in the upper row. We can see that:

- Consistency and redundancy values are no longer more or less uniform across the whole input space. There is a gradient in the values of both criteria corresponding inversely to changes in the temperature. The higher the temperature is, the higher the expected value of the random component is (see Equation (2.21)) and, correspondingly, the lower values of consistency and redundancy are.

- The magnitude of the consistency and redundancy values are inversely proportional to the value of parameter  $K$ . Mean values of consistency and redundancy in Figure 2.18, corresponding to a  $K$  value of 0.05 are clearly higher to those in Figure 2.19, corresponding to a  $K$  value of 0.05. Since parameter  $K$  determines the magnitude of the random part of the error in the ammeter measurements, this confirms the former result that consistency and redundancy values are inversely proportional to the magnitude of the random part of the error.

So, we can deduce a very important result concerning FCFs: *Local consistency and redundancy values are high in zones of the input space with high completeness values and constant, non-random error. Those zones tend to correspond to zones of low error in the output space, that is, zones where the FCF does a good job filtering the ammeter's output. On the other hand, local consistency and redundancy values diminish in zones of the input space with large expected random error (and diminish proportionally to this error). Those zones tend to correspond to zones of high error in the output space, that is, zones where the performance of the FCF filtering the ammeter's output decreases.*

Summarizing, we could compute the confidence we can deposit in the output given by a FCF (not just in the case of the ammeter, but in general) using the following fuzzy algorithm (Comp states for completeness, Cons for consistency and Red for redundancy):

---

**Algorithm 2.1** Confidence computation function
 

---

```

1: function CONFIDENCE(Comp, Cons, Red) : returns confidenceValue;
2:   var confidenceFuzzySet : FuzzySet
3:   confidenceFuzzySet  $\leftarrow$   $\emptyset$ 
4:   if Comp is LOW then
5:     confidenceFuzzySet  $\leftarrow$  confidenceFuzzySet  $\cup$  LOW;
6:   end if
7:   if Comp is HIGH and Cons is LOW and Red is LOW then
8:     confidenceFuzzySet  $\leftarrow$  confidenceFuzzySet  $\cup$  LOW;
9:   end if
10:  if Comp is HIGH and Cons is HIGH and Red is HIGH then
11:    confidenceFuzzySet  $\leftarrow$  confidenceFuzzySet  $\cup$  HIGH;
12:  end if
13:  return defuzzify(confidenceFuzzySet);
14: end function

```

---

Besides the application of FCFs to real world problems (we will see two such applications in the second part of this document), several directions towards further research in the topic can be pointed:

- First, it would be convenient to devise a method to, given the fuzzy system associated to a FCF, determinate whose values of the three criteria, completeness, consistency and redundancy, are considered low or high. Ideally, given values for those three criteria, we would like to know a procedure to deduce or approximate a probability distribution for the expected value of the error committed by the FCF.
- It is not totally clear the relative importance and mutual independence of the three criteria. It is worth to note that, despite the high correlation between consistency and redundancy values in high completeness areas that can be observed in Figures 2.14 to 2.19, they are by no means equivalent. By definition, in areas of low completeness, consistency values will tend to one while redundancy values will tend to zero<sup>8</sup>.
- The choice of those three particular criteria obeys to several practical and theoretical reasons (mainly simplicity and the importance of their crisp counterparts in classical logic). Additional quality criteria (ie. generality, lineality, importance ...) are being studied.
- Application of completeness, consistency and redundancy criteria can be extended to domains other than FCFs. For example, they can be integrated into the fit function of a genetic algorithm used to evolve fuzzy systems.

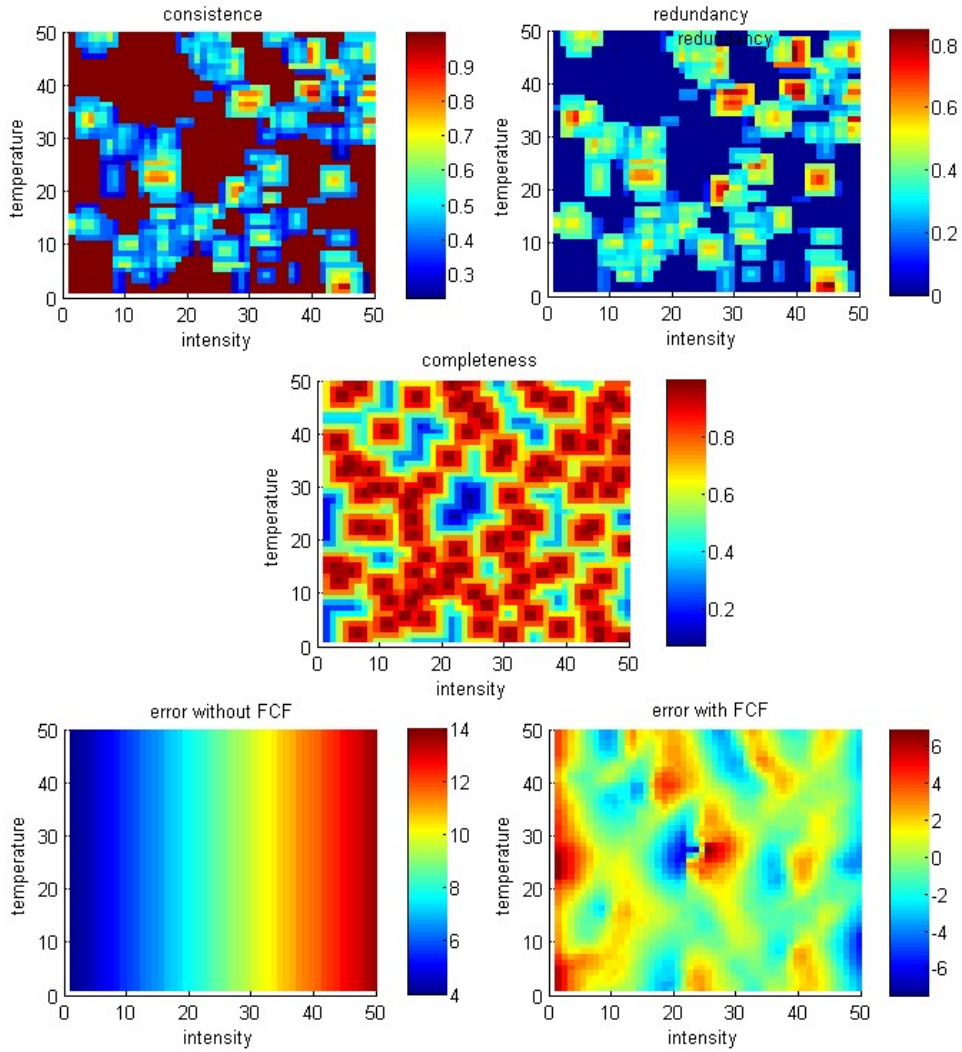
## 2.6 Conclusions

This chapter introduces the novel (concerning the multiagent systems field) concept of reliability. It argues that reliability not only extends the concept of trust, but also beats it in terms of usefulness. Reliability is defined as the quality of an agent that tends to give the same or a similar response when the same or a similar question is asked under the same or similar circumstances, regardless of the erroneous the answer can be. This regularity in the error committed by the agent in his response can allow us to extract knowledge from it.

The device presented in order to make possible the computation of the reliability of a given agent under a given set of circumstances, as well as the

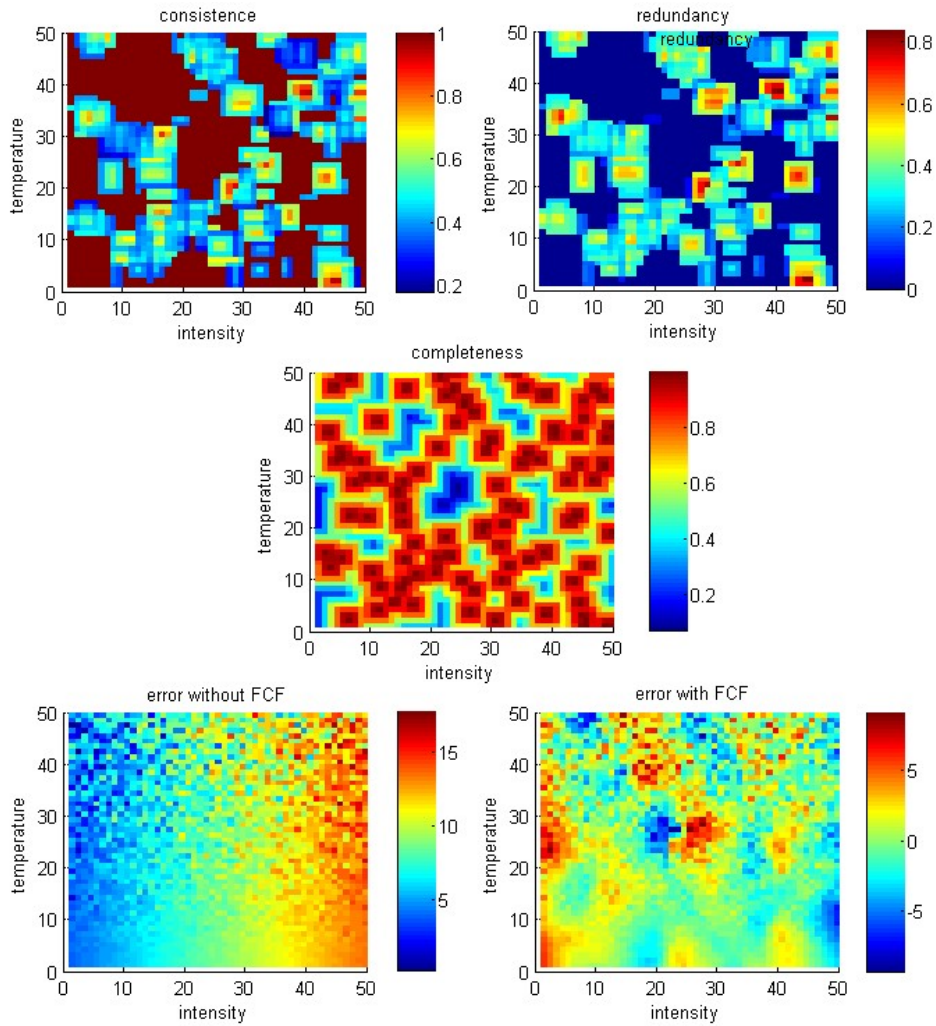
---

<sup>8</sup>So, redundancy is high only in areas of the input space with high completeness and more or less constant error, that is, the zones in the input space where the FCF will perform better. That makes this criteria perhaps the most important of the three.

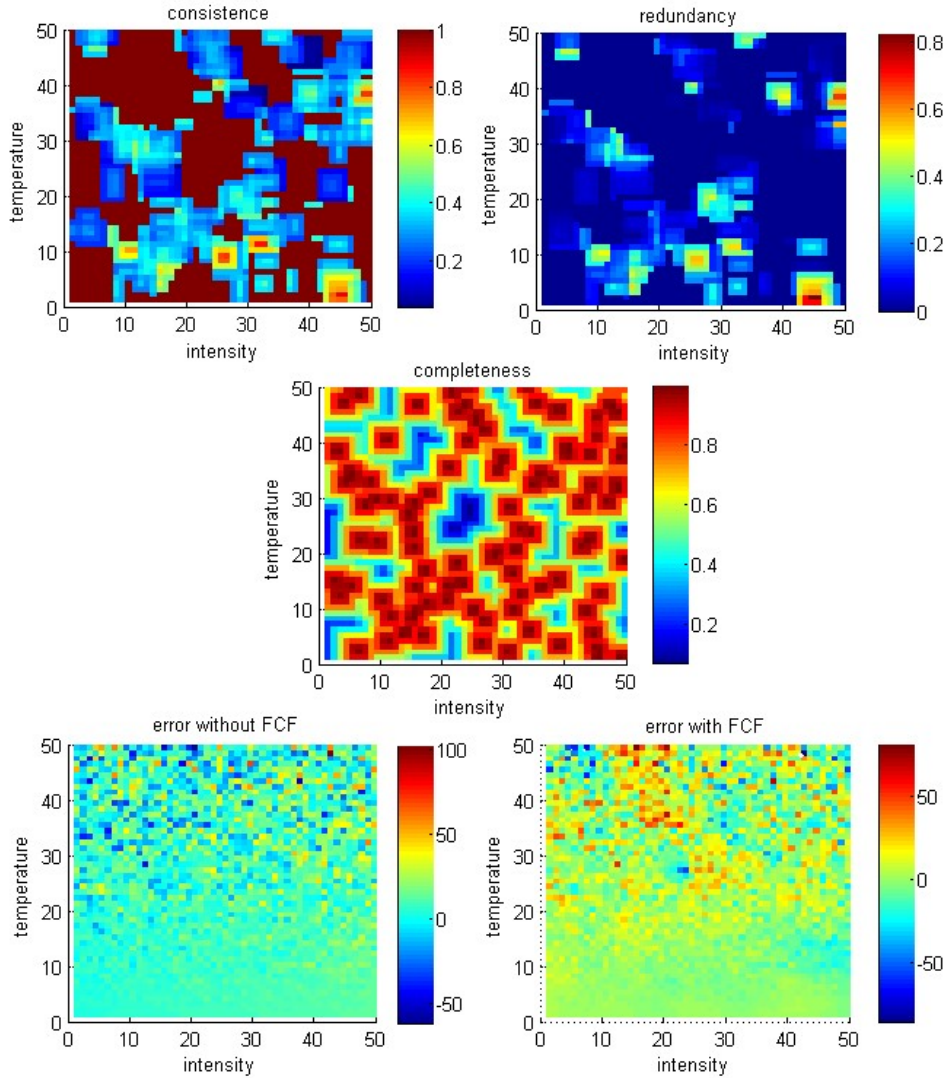


**Figure 2.14:** Top and middle rows: detailed graphs of the local consistency, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 4$ , random error standard deviation  $K = 0$ .

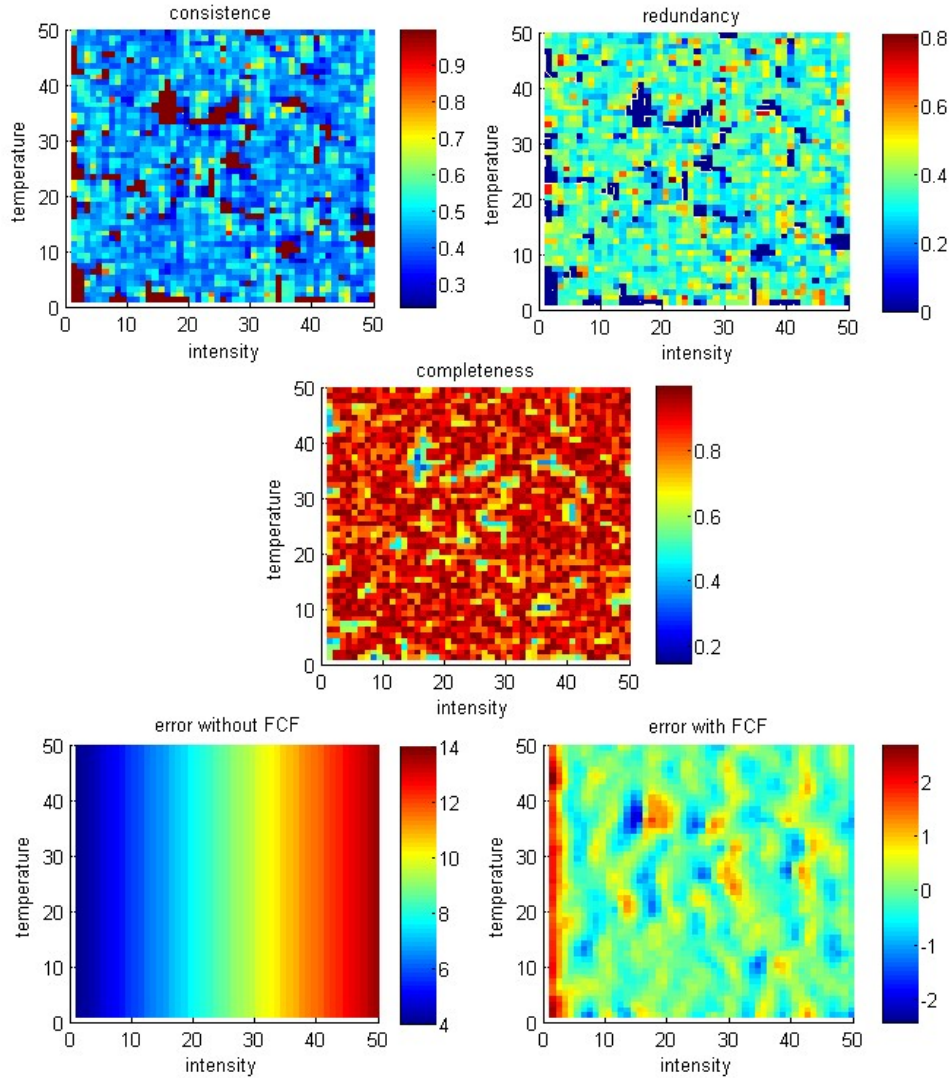




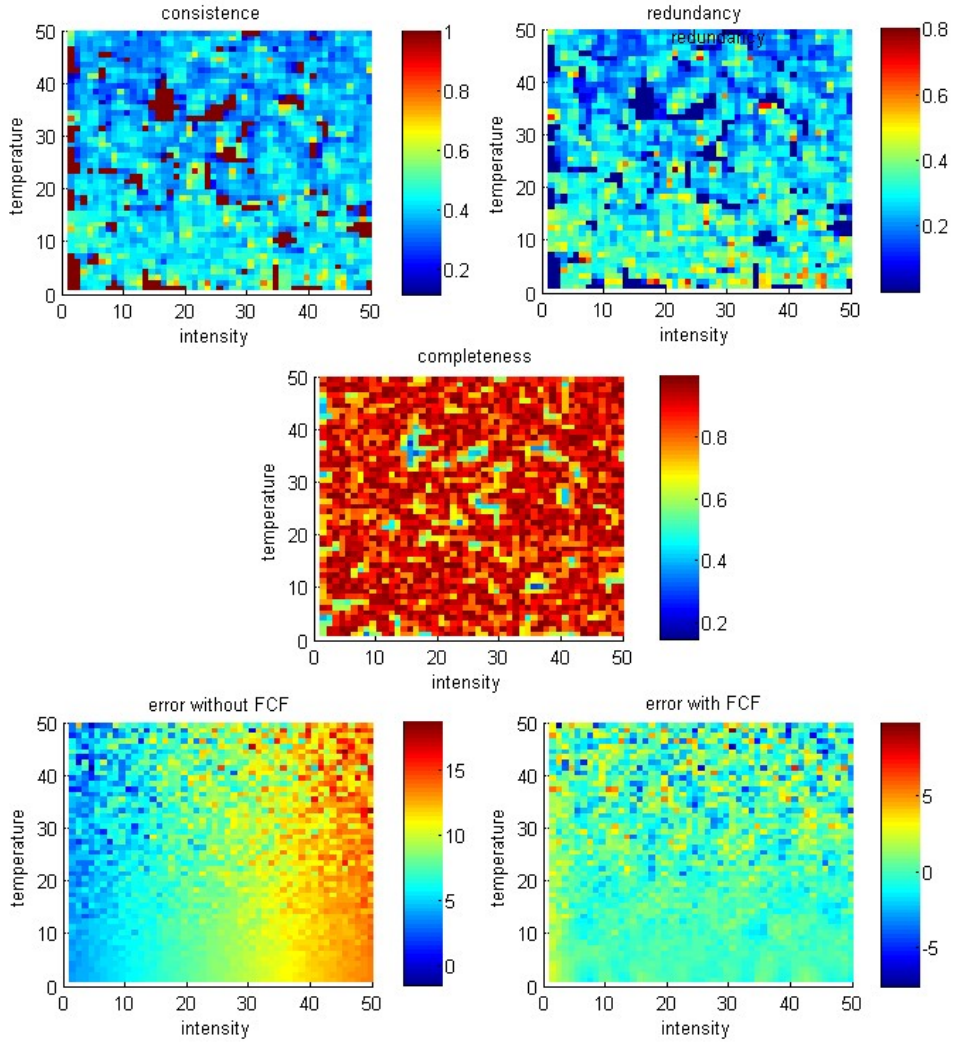
**Figure 2.15:** Top and middle rows: detailed graphs of the local consistency, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 4$ , random error standard deviation  $K = 0.05$ .



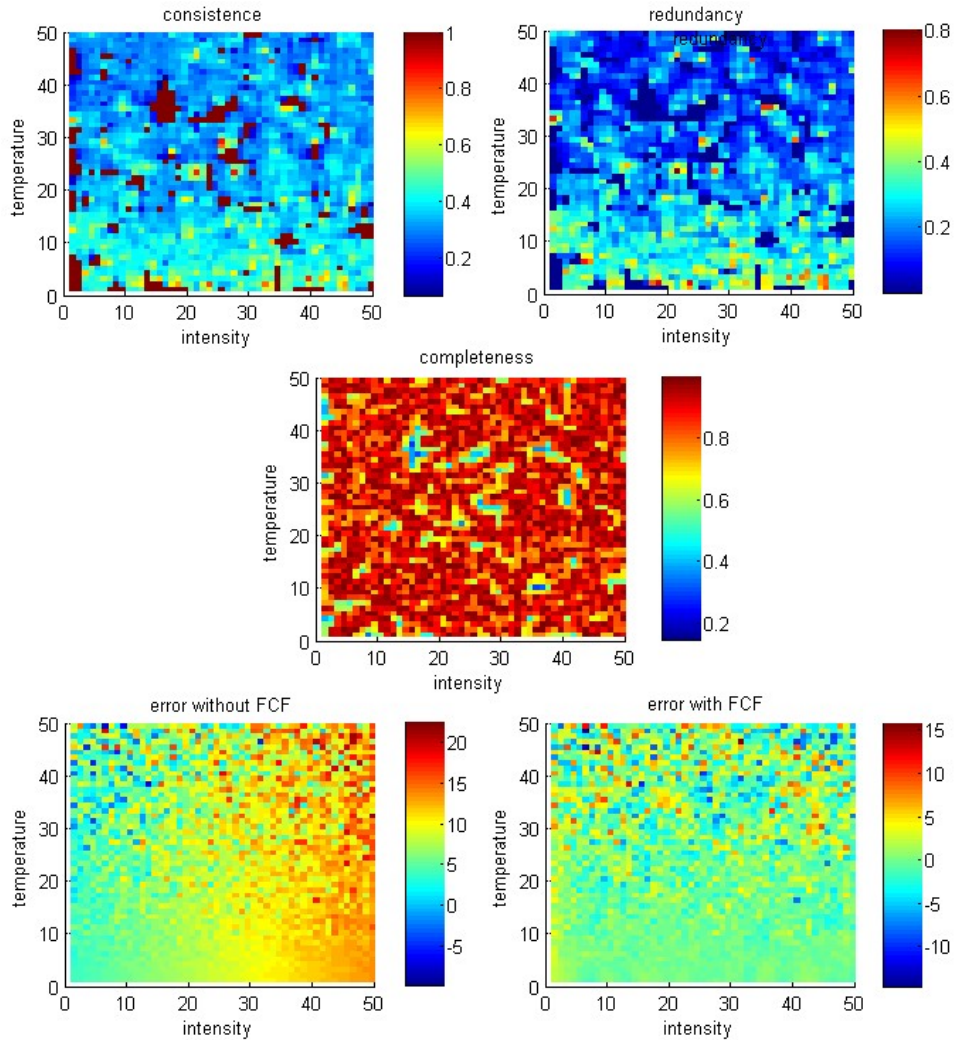
**Figure 2.16:** Top and middle rows: detailed graphs of the local consistency, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 4$ , random error standard deviation  $K = 0.5$ .



**Figure 2.17:** Top and middle rows: detailed graphs of the local consistency, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0$ .



**Figure 2.18:** Top and middle rows: detailed graphs of the local consistency, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0.05$ .



**Figure 2.19:** Top and middle rows: detailed graphs of the local consistency, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0.1$ .

correction of the error committed by the agent, is called a Fuzzy Contextual Filter (FCF). FCFs are composed by two parts: a corrective module, that given an input value provided by an agent and a set of values for other variables in the environment, filters the input value (that is, tries to remove the error from the input value) to obtain a new, filtered value, and a second module, the reliability computation module, which computes the confidence we can deposit in the filtered value provided by the corrective module.

The corrective module of a FCF consists basically in a set of rules forming a fuzzy system. Those rules are obtained from experience or common sense and from past interactions between the agent possessing the FCF and the agent whose output is to be filtered. While the use of a fuzzy system for function approximation is by no means new, the corrective module presents two particular characteristics: In one hand, it incorporates a initial rule base that can express the *a priori* assumptions about the behavior of the other agents in the environment and serve as a departing point in the interpretation of other agents's assertions. On the other hand, the overall structure of the module is fixed, with one main variable (the input variable to be corrected or filtered) and zero or more context variables measuring characteristics in the environment that can affect the error present in the value of the input variable. The output of the module, the filtered variable, is a guess about the actual value of whatever the input variable is meant to measure.

The reliability computation module computes the reliability of the value of the filtered variable obtained by the corrective module. Reliability is a function of the main and context variables, that is, given values for the main and context variables, the reliability calculation module produces a value representing the confidence we can have in the exactitude of the value of the filtered variable computed by the corrective module. Reliability depends upon the number of prior similar interactions between filtering and filtered agents as well as upon the regularities observed during that interactions. Reliability is computed as a combination of three fuzzy system's quality criteria applied to the fuzzy rule bases in the corrective module: completeness, redundancy and consistency. This value can be computed exclusively from the structural characteristics of the system. That is, from the form of its rules and the fuzzy sets used in their definition. The computation of this quality value, then, is easily automatable.

Finally we have described an experiment consisting in a simulation of the behavior of an ammeter under different conditions of input intensity and temperature. The results of the experiment confirm the validity of the approach.

It is interesting to remark that the use of a fuzzy system in the implementation of the corrective module is just a design decision. Given the set of examples, it could be perfectly possible to implement it using a bunch

of different methods ranging from neural nets to support vector machines, passing by wavelets and b-splines. The use of a fuzzy system makes easier the implementation of the reliability module (by means of the definition of several reliability criteria, as completeness, consistency and redundancy). The definition of similar or equivalent criteria in a black box type model, like neural nets, would be much more difficult.

Part II

Applications





## Chapter 3

# Using FCFs for Variance Estimation in the ART Testbed

### 3.1 Introduction

This chapter introduces a first application of Fuzzy Contextual Filters (FCFs) to agent modeling, concretely to the ART (Agent Reputation and Trust) Testbed. It is based upon previous work that can be found in [22]. First sections introduce and discuss in some depth the main aspects and singularities of the ART Testbed. The second part of the chapter presents several of the techniques and algorithms behind the appraiser agent SPARTAN, specially the application of FCFs to the estimation of other agents's appraisals variances. Finally, the chapter includes a discussion of the results of the SPARTAN agent in several national and international championships and make several proposals that, in our opinion, could improve the design of the testbed and its overall use experience.

### 3.2 The ART Testbed

The Agent Reputation and Trust (ART) testbed [3, 32, 33] is a framework, based on the art appraisal domain, for experimentation and comparison of trust modeling techniques. Agents function as painting appraisers with varying levels of expertise in different artistic eras. There is a set of customers (evenly distributed amongst the agents in the beginning of the simulation) who request appraisals for paintings from different eras to the agents, and they have to provide as accurate estimated appraisals as possible for each painting. The clients share, and thus the profit, for the next iteration of the algorithm will depend upon the accuracy of the appraisals of each agent in



Figure 3.1: *ART testbed logo.*

the previous iteration. When an appraising agent believes that he does not have expertise enough to produce a good appraisal for a given painting, he can request opinions from other, hopefully more experts, appraiser agents.

### 3.2.1 Client Appraisals

In each timestep, multiple clients present each appraiser (agent) with paintings to be appraised, paying a fixed fee  $F$  for each appraisal request. To increase business, appraisers attempt to value paintings as closely to market value as possible. A given painting may belong to any of a finite set of eras (a painting’s era is known by all appraisers), and appraisers have varying levels of expertise in each era. An appraiser’s expertise, defined as its ability to generate an opinion about the value of a painting, is described by a normal distribution of the error between the appraiser’s opinion and the true painting value. The simulation creates opinions according to this error distribution, which has a mean of zero and a standard deviation  $s$  given by

$$s = \left( s^* + \frac{\alpha}{c_g} \right) t \quad (3.1)$$

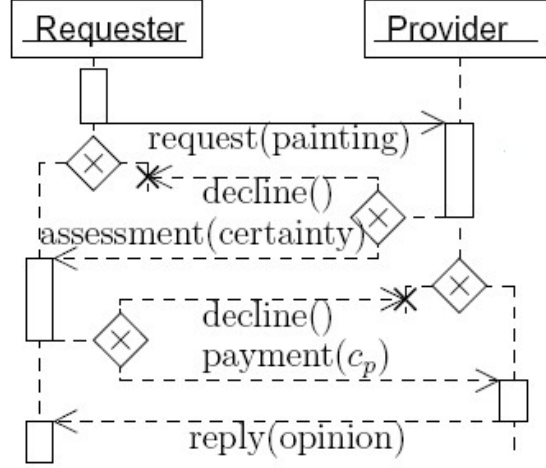
where  $s^*$ , unique for each era, is assigned to an appraiser from a uniform distribution in  $[0 \cdots 1]$ ,  $t$  is the true value of the painting to be appraised and  $\alpha$  is a parameter, chosen by the experimenter and fixed for all appraisers, which affects the relationship between opinion-generation cost and resulting accuracy. Parameter  $c_g$ , the cost an appraiser is willing to pay to generate an opinion, is discussed in more detail below. An appraiser’s expertise for each era does not change throughout the duration of a game, and appraisers know their levels of expertise for each era. However, the simulation does not inform appraisers of other appraisers’s expertise levels. The true values of

paintings presented by clients are chosen from a uniform distribution known only to the simulation; likewise, the eras to which paintings belong are also uniformly distributed among the set of eras. An appraiser chooses a variable cost  $c_g$ , representing time taken to examine the painting, to pay in generating its own opinion about a painting's value. An appraiser is required to pay a minimum  $c_g$  of one monetary unit. By paying a higher  $c_g$ , analogous to spending more time studying the painting, an appraiser increases the accuracy of its opinion. However, an appraiser cannot perfectly judge a painting by spending an infinite amount of time studying it; the appraiser's accuracy is still limited by its expertise. The minimum achievable error distribution standard deviation is  $s^* + t$ . In addition to generating its own opinion, an appraiser may request opinions from other appraisers to improve its final appraisal. This is especially important when an appraiser attempts to value paintings from eras for which it has low expertise.

Appraisers may request opinions from as many other appraisers as desired for each painting, at a fixed cost  $c_p$  for each opinion transaction. In general,  $c_p \ll f$  (where  $f$  is the price a client pays for an appraisal) to encourage opinion exchange. Appraisers may also provide opinions for as many paintings as desired in a single time step. Appraisers are not required to truthfully reveal their opinions; they can communicate false opinions if desired, for example, in attempting to decrease the requester's client base and resulting profit. The simulation oversees each portion of a timestep synchronously, including client requests, appraiser opinion generation, transactions between appraisers, and returning final appraisals to clients. Therefore, appraisers are required to perform required actions for each time step portion within real-time limitations as monitored by the simulation.

### 3.2.2 Opinion Transactions

The opinion transaction protocol is shown in Figure 3.2. To initiate an opinion transaction, a requester sends a request message to another appraiser (potential opinion provider), identifying the painting to be appraised. Upon receiving an opinion request, if the potential provider is willing to provide the requested opinion, it responds by sending a certainty assessment about the opinion it can provide, defined as a real number between zero and one (one represents complete asserted certainty). The potential provider is not required to provide a truthful certainty assessment. If the potential provider does not wish to participate in the requested transaction, it may choose to decline the request. By sending a certainty assessment, the provider promises to deliver the requested opinion should the certainty assessment be accepted by the requester. After receiving the provider's certainty assessment, the requester either sends payment to the provider if it chooses to accept the promised opinion, or sends a 'decline' message if it chooses not to continue



**Figure 3.2:** *The opinion transaction protocol (from [32]).*

the transaction. The cost of each transaction is the non-negotiable amount  $c_p$ . Upon receipt of payment, the provider is not required to send its actual opinion, neither is the provider forced to send any opinion at all. Upon paying providers, but before receiving opinions from providers, the requesting appraiser is required to submit to the simulation its roster of opinion providers and a set of corresponding weights. Weights are values between zero and one, loosely representing the appraiser’s confidence or trust in each provider’s opinion.

The appraiser’s final appraisal  $p^*$  is calculated by the simulation as a weighted average of received opinions:

$$p^* = \frac{\sum_i (w_i \cdot p_i)}{\sum_i (w_i)} \quad (3.2)$$

where  $w_i$  and  $p_i$  are the appraiser’s weight for, and the received opinion from each provider  $i$  whose opinion was received (possibly including itself). The true painting value  $t$ , along with the calculated final appraisal  $p^*$ , is then revealed by the simulation to the appraiser. The simulation enforces this roster submission and final appraisal calculation protocol; requesting appraisers are not permitted to change their rosters or alter received opinions from providers. The simulation calculates final appraisals to prevent appraisers from developing non-trust-based appraisal calculation strategies and allow appraisers to focus on the more important task of assessing and selecting trustworthy opinion providers. Upon learning its final appraisal and the painting’s true value, an appraiser may use this feedback to revise its trust models of other appraisers.

Each appraiser has a bank account, monitored by the simulator, from which it pays transaction costs and into which are deposited client appraisal fees. Bank accounts are initialized with zero and may hold negative balances. Competing appraisers can not observe each other’s clients or bank account balances.

### 3.2.3 Reputation in the ART Testbed

In addition to conducting opinion transactions, and following approximately the same protocol, appraisers can exchange information about the trustworthiness of other appraisers, that is, their reputation. So, an agent, before deciding the buying of one appraisal from a given appraiser agent, can ask other agents their opinion about its honesty and ability.

Unfortunately, reputation consulting has not much interest in the current (2007) form of competition/simulation in the ART testbed. Due to implementation limitations, the maximum allowed number of agents per simulation is very low (five or six, typically). This causes every agent to quite frequently interact with each other, thus having the opportunity to build accurate models. In this situation, it does not make much sense to spend money buying (probably false) opinions from other agents.

This issue is going to be fixed in upcoming versions of the simulator, which will allow hundreds of agents to participate simultaneously in the simulations. In this new scenario, we firmly believe that the interchange of information about reputation will have the greatest impact in agent’s performance. In the meanwhile, no reputation interchanges were used in the current implementation of the SPARTAN agent. That decision has been shared by the practical totality of other competitors in the ART championships.

### 3.2.4 Client Share Assignment

Although clients are initially evenly distributed among appraisers, appraisers whose final appraisals are most accurate are rewarded with a larger share of the client base in subsequent timesteps. To calculate each appraiser’s share of the client base for the next simulation, each appraiser  $a$ ’s average relative appraisal error for the present iteration,  $\epsilon_a$  is first calculated (adapting the formula from [32]):

$$\epsilon_a = \frac{\sum_{q \in T_a} \frac{|q.AppVal - q.TrueVal|}{q.TrueVal}}{|T_a|} \quad (3.3)$$

where  $T_a$  is a set of triplets in the form  $(Painting, AppVal, TrueVal)$  representing the set of appraisals made by  $a$ . For each appraisal  $q \in T_a$ ,  $q.AppVal$  and  $q.TrueVal$  represent respectively the value by which  $q.Painting$  was appraised and its true value.

Next, each appraiser  $a$  is assigned a preliminary client share for the next iteration  $\bar{r}_a$  according to its average relative appraisal error:

$$\bar{r}_a = \left( \frac{\delta_a}{\sum_{b \in A} \delta_b} \right) \cdot |C| \quad (3.4)$$

where  $A$  is the set of all appraisers,  $C$  is the set of all the clients (or paintings to be appraised) and

$$\delta_a = 1 - \frac{\epsilon_a}{\sum_{b \in A} \epsilon_b} \quad (3.5)$$

Thus, the appraiser with the least average relative appraisal error achieves the highest preliminary client share for the next iteration.

Finally, each appraiser  $a$ 's final client share for the next iteration  $r_a$  is computed as:

$$r_a = q \cdot r'_a + (1 - q) \cdot \bar{r}_a \quad (3.6)$$

where  $r'_a$  has been appraiser  $a$ 's final client share in the present iteration. The parameter  $q$ , a value between zero and one inclusive, reflects the influence of previous client share size on next client share size. Thus the volatility in client share magnitudes due to frequent accuracy oscillations is reduced for larger values of  $q$ , which is chosen by the experimenter and is the same for calculating all appraisers' client shares.

### 3.3 Several Considerations on the Client Share Assignment Function

As we have just seen, ART designers propose, in [32, 33] the following equation for client share adjustment for agent  $a$  after each iteration

$$r_a = q \cdot r'_a + (1 - q) \cdot \bar{r}_a \quad (3.7)$$

where  $r'_a$  is agent  $a$ 's client share in the previous timestep,  $q$  is a parameter whose value lies in the  $[0 \dots 1]$  interval and  $\bar{r}_a$ , depending on the mean relative error committed by  $a$  (as well as by the other agents), represents  $a$ 's preliminary client share for the current iteration. The formula for the computation of  $\bar{r}_a$  is:

$$\bar{r}_a = \left( \frac{\delta_a}{\sum_{b \in A} \delta_b} \right) \cdot |C| \quad (3.8)$$

where  $C$  is the set of customers or paintings to be assigned in the current iteration and  $\delta_a$  represents the relative quality of  $a$ 's appraisals in the previous

time step.  $\delta_a$  is computed as:

$$\delta_a = 1 - \frac{\epsilon_a}{\sum_{b \in A} \epsilon_b} = \frac{\sum_{b \in A} \epsilon_b - \epsilon_a}{\sum_{b \in A} \epsilon_b} \quad (3.9)$$

representing  $\epsilon_b$  the mean relative error made by agent  $b$  during the previous iteration. Substituting  $\delta_a$ 's values in equation 3.8 and simplifying we have:

$$\bar{r}_a = \frac{\sum_{b \in A} \epsilon_b - \epsilon_a}{\sum_{b \in A} (\sum_{c \in A} \epsilon_c - \epsilon_b)} \cdot |C| = \frac{\sum_{b \in A, b \neq a} \epsilon_b}{\sum_{b \in A} \sum_{c \in A, c \neq b} \epsilon_c} \cdot |C| \quad (3.10)$$

this can easily be proved equivalent to:

$$\bar{r}_a = \frac{|C|}{|A| - 1} \cdot \frac{\sum_{b \in A, b \neq a} \epsilon_b}{\sum_{b \in A} \epsilon_b} \quad (3.11)$$

or, alternatively

$$\bar{r}_a = \frac{|C|}{|A| - 1} \cdot \frac{\Sigma}{\epsilon_a + \Sigma} \quad (3.12)$$

where, as stated before and following the notation in [32],  $\epsilon_b$  represent the mean relative error made by agent  $b$  during the previous iteration,  $A$  and  $C$  are respectively the sets of agents and clients and  $\Sigma$  is simply the sum of the mean relative errors made by the agents other than  $a$ .

It is worth to make a couple of considerations about equations 3.11 and 3.12. Firstly, it is clear from the equations that no agent can manage to get a preliminary client share greater than  $|C|/(|A| - 1)$ , never mind the accurateness of its appraisals. An agent in a seven-agent environment, for example, can't aspire to obtain more than one sixth of the total client amount, even if it guesses exactly all the pictures' prices and the other agents make huge errors in their appraisals. On the contrary, a big mean relative error can dramatically diminish the preliminary client share of an agent to one single client, in the worst case (not to zero because of ART's design, see [32]). This is critical due to the fact that it is possible for an agent to "fool" another one inducing a huge error in one of its appraisals (by spending a very little amount of money in its appraisal creation and therefore increasing the appraisal error of the standard deviation, see Equation 3.1.)<sup>1</sup>. So, we could

---

<sup>1</sup>It has not been unusual in some simulations to see agents with relative mean errors of such a magnitude as  $10^{20}$ . Such a large error will completely dominate Equation 3.1 and will make irrelevant the amount of money spent by the other agents in the creation of their appraisals and whether their errors are small or big. All of them will receive quite the same preliminary client share in the next iteration. We can not avoid looking at this issue as a shortcoming in the design of the platform.



say that ART's preliminary client share assignment method penalizes the bad appraisals much more than it rewards the good ones. It will be important to take this into account when designing our appraising agent strategy, it seems to make much more sense to spend money trying to reduce the relative error in the cases when we suspect it to be large than refining opinions we can suppose to be quite accurate. On the same line, benefits from deceiving an agent by selling a deliberately wrong appraisal to it (thus causing it to lose a big share of its customers for the next iteration) are equally shared amongst the deceiver agent and all the other agents. This can make deceiving tactics counterproductive (mainly in populated environments, where there are more agents to share the customers lost by the deceived agent) due to the loss of reputation not being compensated by the difference in earnings. Perhaps it would be interesting to study how robustly current agents would behave if a more deception-encouraging formula for client assignment was used (something like the deceiver agent keeping all the customers lost by the deceived agent, for example).

The second consideration to be made refers to the fact that computation of the preliminary client share for agent  $a$  takes not into account the actual distribution of the relative error amongst the other agents, but the total amount of this error. So, an agent can, at the beginning of each iteration, knowing the mean relative error made by itself in the last iteration and the client share assigned to it in the current one, use Equations 3.7 and 3.12 to compute the total mean error made for all the remaining agents in the past iteration and use this value as a estimator of the total mean error which the remaining agents will made in the next iteration. This will help the agent to establish a near-optimal strategy regarding the amount of money it has to spend refining its own appraisals as well as purchasing appraisals from other agents in the current iteration. We will come back to this in section 3.9.

### 3.4 The SPARTAN Algorithm

Algorithm 3.1 describes in general terms the behavior of the SPARTAN agent every time it has to appraise a painting  $P$ . First, the agent asks each other agent in the simulation the confidence it has in its ability to correctly appraise painting  $P$ . So it ends up this step having a set of values between zero and one representing different degrees of confidence and having different meanings for each agent (that is, a confidence value of 0.5 does not means the same thing for every agent). In order to solve this problem and make possible to use consistently these values, SPARTAN has to translate them to some common concept it can use to combine the appraisals effectively. We know that the expectation of the error for each agent appraisal is zero (by design). This means that what determines the accuracy of the appraisals is the expectation

of the error's variance, so it makes sense, in the step 2 of the algorithm, to try to translate the confidence values in some way such that we could estimate that expectation. In order to accomplish this accurately, SPARTAN will need a set of models of the other agents. Those models (we will use FCFs to implement them, more on this in section 3.8) will be built iteratively along the history of successive interactions. Once it has an estimation of the error, SPARTAN will compute a suitable value for  $c_g$  (the amount of money it will invest in its own appraisal), depending mainly in the expertise it has in painting  $P$ 's era. Next, SPARTAN decides the set of agents from which it will purchase appraisals and computes the vector of weights to be transmitted to the simulator. Finally, when the appraisal is over, SPARTAN asks the simulator about the true value of the painting  $P$  and uses the information to update the models of the other agents.

---

**Algorithm 3.1** General appraisal algorithm for the SPARTAN agent.

---

- 1: **procedure** APPRAISE(P:Painting)
  - 2:     **Ask** every agent their appraisal confidence for P
  - 3:     **Compute** the variance of the relative error for each agent
  - 4:     **Compute** the best value for  $c_g$
  - 5:     **Select** a set of agents to which purchase appraisals
  - 6:     **Form** the weight vector according to the computed variances and transmit it to the simulator
  - 7:     **Ask** the real price of the painting to the simulator
  - 8:     **Update** the information about the agents
  - 9: **end procedure**
- 

There are left quite a few important questions to answer and also several main design issues that have to be solved in the implementation. There are:

1. How to translate accurately the confidence values provided by the agents to expected relative errors in their appraisals?
2. How to translate, in case another agent asks it, SPARTAN expected relative error for an appraisal to a confidence value?
3. How to compute a good value for  $c_g$ ?
4. How to select the subset of agents to purchase an appraisal from?
5. How to form the optimal weights vector, given a set of expected variances for the appraisal error?
6. How to update SPARTAN models of the other agents?

We will discuss them in the following sections (although not in the same order).

### 3.5 The Taxation Vectors

Let’s introduce the main data structures SPARTAN will use and a little bit of notation. We will have, for each agent  $a$ , a **taxation vector**  $R_a$  where it will store information relating to all the taxations it has purchased from agent  $a$ . It will fill up this vector at the end of each iteration. Each element  $R_a[i]$  will have three fields:

- $R_a[i].TimeStep$ . The simulation time step in which the taxation took place.
- $R_a[i].ConfidenceVal$ . The confidence manifested by the agent  $a$ .
- $R_a[i].QuadRelError$ . The square of the relative error done by agent  $a$  in its taxation. We define it as:

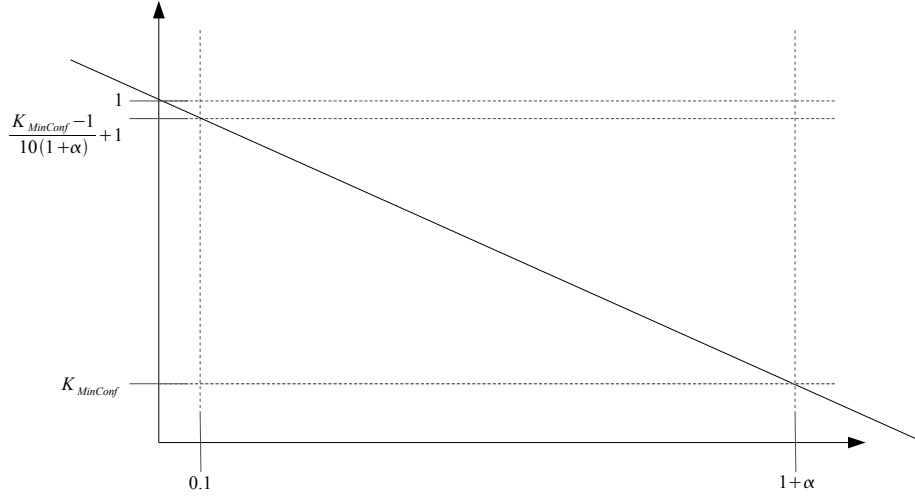
$$R_a[i].QuadRelError = \left(1 - \frac{taxatedValue_i}{realValue_i}\right)^2 \quad (3.13)$$

The reason to store the square of the relative error instead of the relative error itself (or even the absolute error) is to ease the calculations of the variance (and standard deviations) of the relative errors in agents’s appraisals. We will go again into it in section 3.8.

It can be observed that we don’t store in the taxations vector the epochs corresponding to the appraised paintings. We will assume that agents “talk” the same language independently of the epoch the painting belongs to, that is, we will assume that a given confidence value provided by an appraiser agent will mean the same thing when referred to whichever epoch. It seems a sensible assumption which will allow us to quickly recollect information about appraiser agents’ behavior and, consequently, to build more accurate models of the agents at the earlier steps of the simulation.

### 3.6 Translating Standard Deviation of Relative Errors to Confidence Values

When SPARTAN is asked to provide a confidence for the taxation of a painting of a given era, it has to take into account its expertise in the painting’s era as well as the amount of money it is willing to spend in the generation of its appraisal to respond with a value in the  $[0 \dots 1]$  interval representing its confidence in the accuracy of its upcoming appraisal. SPARTAN is a honest agent, so the method we will use will be honest and straightforward. We will make depend this confidence value lineally upon the standard deviation of the relative error, that we can compute from Eq. 3.1 by dividing it by the



**Figure 3.3:** Plot of the *conf* function

real price of the painting. The value of the standard deviation of the relative error, then, will be given by:

$$\bar{s} = \left( s^* + \frac{\alpha}{c_g} \right) \quad (3.14)$$

where, according to ART testbed designers [32], the value of the inherent standard deviation  $s^*$  lies in the interval  $[0.1 \dots 1]$  and the value of the parameter  $\alpha$  is fixed for all the agents and eras. So, (assuming  $c_g \geq 1$ , that is, we spend at least one money unit in the taxation-making process) the value of  $\bar{s}$  will range between 0.1 and  $1 + \alpha$  and we can define the confidence associated to the value  $\bar{s}$  of the standard deviation of the relative error by means of a simple linear function:

$$conf(s) = \frac{K_{MinConf} - 1}{1 + \alpha} \cdot s + 1 \quad (3.15)$$

Figure 3.3 shows a graph of the *conf* function. We have chosen not to allow the *conf* function to take values over the whole  $[0 \dots 1]$  interval, so a maximal  $(1 + \alpha)$  standard deviation of the relative error corresponds to a constant minimal confidence  $K_{MinConf}$  and a minimal (0.1) standard deviation of the relative error does not correspond to the unity value (which would mean full confidence, that is, no error).

### 3.6.1 Translating Confidence Values to Standard Deviations

Alternatively, we don't know "a priori" the meaning the other agents intend to give to their statements about the confidence they have in their opinions.

We need to interpret those statements. Concretely, we need a way to estimate the (standard deviation of the) relative error which we can expect from the other agents, departing from the confidence values they have provided. As the number of interactions with those agents increase, we will be able of making more accurate interpretations, but in the meanwhile, we need a method for doing those translations from the beginning of the auction process, we will do it by means of the inverse of the *conf* function:

$$\text{conf}^{-1}(c) = (c - 1) \cdot \frac{1 + \alpha}{K_{\text{MinConf}} - 1} \quad (3.16)$$

where  $c$  stands for the confidence value provided by the consulted agent. It is worth to stress the fact that we wont rely too heavily in the anterior function in order to interpret other agents' assertions, we will use it mainly as a sensible starting point for bootstrapping purposes when little o no further information about the other agents is available. We will see in section 3.8 how to make use of the information gathered in subsequents interactions with the agents to translate more accurately the confidence values provided by them.

### 3.7 Determining the Optimal Set of Weights in Appraisal Combination

Following [32], in the ART testbed the appraisal error is distributed as a normal aleatory variable with mean 0 and standard deviation:

$$s = \left( s^* + \frac{\alpha}{c_g} \right) \cdot t \quad (3.17)$$

where  $s^*$  is the inherent standard deviation for the epoch to which the painting belongs,  $\alpha$  is a simulation dependent constant and  $c_g$  is the amount of money spent in the appraising process. Thus, the standard deviation for the appraisals will also be  $s$  and the standard deviation of the relative errors defined as  $\frac{\text{appraise}-t}{t}$  will be:

$$\sigma = \frac{s}{t} = \left( s^* + \frac{\alpha}{c_g} \right) \quad (3.18)$$

We will now study the following problem: given a set  $A = \{a_1, a_2 \dots a_n\}$  of appraisals for a painting and given the set  $V = \{\sigma_1^2, \sigma_2^2 \dots \sigma_n^2\}$  of the corresponding variances of the relative errors (we assume that the means of the relative errors of the appraisals are all zero), we want to find the set of weights  $W = \{w_1, w_2 \dots w_n\}$  such that  $w_i \geq 0$  and  $\sum_i w_i = 1$  which minimizes the error of the combined appraisal  $\bar{t} = \sum_i w_i \cdot a_i$ . We will do it by minimizing the variance of the combined appraisal  $\bar{t}$ .

Let's begin with the case of two appraisals  $a$  and  $b$ , with known variances of the relative errors  $\sigma_a^2$  and  $\sigma_b^2$ , we can compute the variance of the appraisal  $\bar{t} = p \cdot a + (1 - p) \cdot b$  as follows:

$$\begin{aligned} Var(\bar{t}) &= Var(p \cdot a + (1 - p) \cdot b) \\ &= p^2 \cdot Var(a) + (1 - p)^2 \cdot Var(b) \end{aligned} \tag{3.19}$$

where  $Var(a)$  and  $Var(b)$  are the variances of the appraisals. From Eqs. 3.17 and 3.18, and taking into account that appraisals have the same variance than errors, we can express  $Var(a)$  and  $Var(b)$  in terms of the known variances of the relative error  $\sigma_a^2$  and  $\sigma_b^2$  using the following equivalence:

$$Var(a) = t^2 \cdot \sigma_a^2 \tag{3.20}$$

where  $t$  is the real value of the painting. We can then restate equation 3.19 as follows:

$$Var(\bar{t}) = t^2 (p^2 \cdot \sigma_a^2 + (1 - p)^2 \cdot \sigma_b^2) \tag{3.21}$$

We can now differentiate:

$$\frac{d(Var(\bar{t}))}{dp} = 2 \cdot t^2 (p \cdot \sigma_a^2 - (1 - p) \cdot \sigma_b^2) \tag{3.22}$$

Equalling to 0 and solving for  $p$ , we have

$$p = \frac{\sigma_b^2}{\sigma_a^2 + \sigma_b^2} \tag{3.23}$$

So, the appraisal of minimal variance will be:

$$\bar{t} = \frac{\sigma_b^2}{\sigma_a^2 + \sigma_b^2} \cdot a + \frac{\sigma_a^2}{\sigma_a^2 + \sigma_b^2} \cdot b \tag{3.24}$$

Finally, the variance of  $\bar{t}$ 's relative error,  $\sigma_{\bar{t}}^2$  can be obtained from equations 3.20 and 3.21 as:

$$\sigma_{\bar{t}}^2 = \frac{\sigma_a^2 \cdot \sigma_b^2}{\sigma_a^2 + \sigma_b^2} \tag{3.25}$$

From this, it is easily proved that, as expected, the optimal combination of appraisals makes appraisal variance lessen, that is:  $\sigma_{\bar{t}}^2 \leq \sigma_a^2, \sigma_b^2$

Those previous results can be easily generalized to the case of a greater number of appraisals. In the case of three appraisals  $a, b$  and  $c$ , the minimal variance appraisal will be:

$$\bar{t} = \frac{\sigma_b^2 \sigma_c^2 \cdot a + \sigma_a^2 \sigma_c^2 \cdot b + \sigma_a^2 \sigma_b^2 \cdot c}{\sigma_a^2 \sigma_b^2 + \sigma_b^2 \sigma_c^2 + \sigma_a^2 \sigma_c^2} \tag{3.26}$$

with a variance of  $\bar{t}$ 's relative error

$$\sigma_{\bar{t}}^2 = \frac{\sigma_a^2 \cdot \sigma_b^2 \cdot \sigma_c^2}{\sigma_a^2 \sigma_b^2 + \sigma_b^2 \sigma_c^2 + \sigma_a^2 \sigma_c^2} \quad (3.27)$$

and, in the general case, if we have a set  $A = \{a_1, a_2, \dots, a_n\}$  of appraisals and the set  $V = \{\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2\}$  of the corresponding relative error variances:

$$\bar{t} = \frac{\sum_{i=1}^n a_i \cdot \prod_{j \neq i} \sigma_j^2}{\sum_{i=1}^n \prod_{j \neq i} \sigma_j^2} \quad (3.28)$$

with a variance of  $\bar{t}$ 's relative error

$$\sigma_{\bar{t}}^2 = \frac{\prod_{i=1}^n \sigma_i^2}{\sum_{i=1}^n \prod_{j \neq i} \sigma_j^2} \quad (3.29)$$

### 3.8 Variance Estimation Using Fuzzy Contextual Filters

When an agent A does not have expertise enough to guarantee a good appraisal for a given painting, it can buy the opinion of other, more expert, agents. The process is the following: first, agent  $a$  asks all or part of the other agents to provide a value stating their confidence in the accuracy of their appraisal of the painting. Then,  $a$  decides, upon the received confidence values, which agents to trust, that is, which opinions to purchase. This is the main point where the communication of false or misleading information can happen in the ART testbed. An agent can declare a great confidence in its appraisal just to fool the requesting agent into purchasing it, and then produce a very bad appraisal (by spending a very little amount of money in its appraisal creation and therefore increasing the appraisal error of the standard deviation, as stated in Section 3.3, see Equation 3.1. This will result in a big error in the requesting agent's appraisal and, consequently, a big loss in its the client share. On the other hand, the requesting agent has no way to know what the confidence value provided by an agent means. It is a value over an arbitrary range that has to be interpreted. It is perfectly possible for a given confidence value to mean completely different confidence levels for different agents.

Suppose that one agent has decided to buy appraisals from one or more other agents in order to produce (taking also into account its own opinion) its final appraisal for a painting. We have just seen in Section 3.7 how to combine optimally two or more appraisals, given the variances of their relative errors, in order to obtain the appraisal with the minimal expected relative error. The drawback is: the agent does not know *a priori* what the variances of the

appraisals to be combined are<sup>2</sup>. The agent needs, therefore, a way to guess these variances from the confidence values supplied by the appraise-selling agents regarding to the painting to be appraised. The agent can make his guess more accurate taking into account the history of past interactions with them.

The problem can be solved by providing the agent with a set of FCFs, one for each agent other than itself in the environment. Those FCFs will “translate” confidence values to expected variances of the relative errors for each agent’s appraisal. The structure of each filter will be very simple. They will have, as input variable, the confidence value stated by the appraisal-selling agent, and, as the filtered variable the expected variance of the appraisal-selling agent’s appraisal relative error. The rule bases of the FCFs will be populated throughout the simulation from the consecutive interactions between agent  $a$  and the other agents. The FCFs, then, will produce their output from the confidence value provided by the appraisal-selling agents and the information collected about them in past interactions.

Let’s see the structure of rule bases in the corrective module of each FCF: rules in the initial rule base are predefined by design implementing Equation 3.16 and serve the purpose of providing a sensible starting point to the interpretation process. Rules in the dynamic rule base, on the other hand, will be continuously obtained from interactions between our agent and the appraisal-selling agent. Each of the rules in the global rule base  $R$  (the union of initial and dynamic rule bases) has the same form:

$$R_i : \mathbf{If} \textit{ ConfidenceVal} = C_i \mathbf{ then} \textit{ QuadRelError} = E_i$$

where  $E_i$  is a singleton fuzzy set over the set of the reals and  $C_i$  is a fuzzy real number. So, for instance, if we purchase an appraisal for a cubist painting for which the appraisal-selling agent declares to have a confidence 0.3, and the provided appraised value is 30000 but the real price of the painting turns out to be 20000 (giving a relative error of 0.5), we will add to our dynamic rule base the following rule (see Figure 3.4):

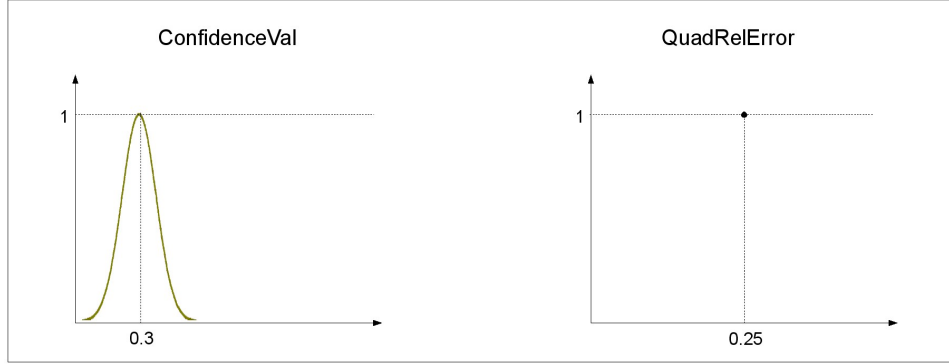
$$\mathbf{If} \textit{ ConfidenceVal} = 0.3 \mathbf{ then} \textit{ QuadRelError} = 0.25$$

We will have, then, a possibly large number of fuzzy rules in this form. Now suppose that agent  $a$  wants to consider the possibility of purchasing an appraisal for a painting from another agent  $b$  which states that it has a confidence  $C$  in its appraisal. How to estimate the variance of the relative

---

<sup>2</sup>That is, the appraisals corresponding to the other agents. The agent knows what the variance of the error in its own appraisal is.





**Figure 3.4:** Fuzzy sets corresponding to the rule : **If** *ConfidenceVal* = 0.5 **then** *QuadRelError* = 0.25

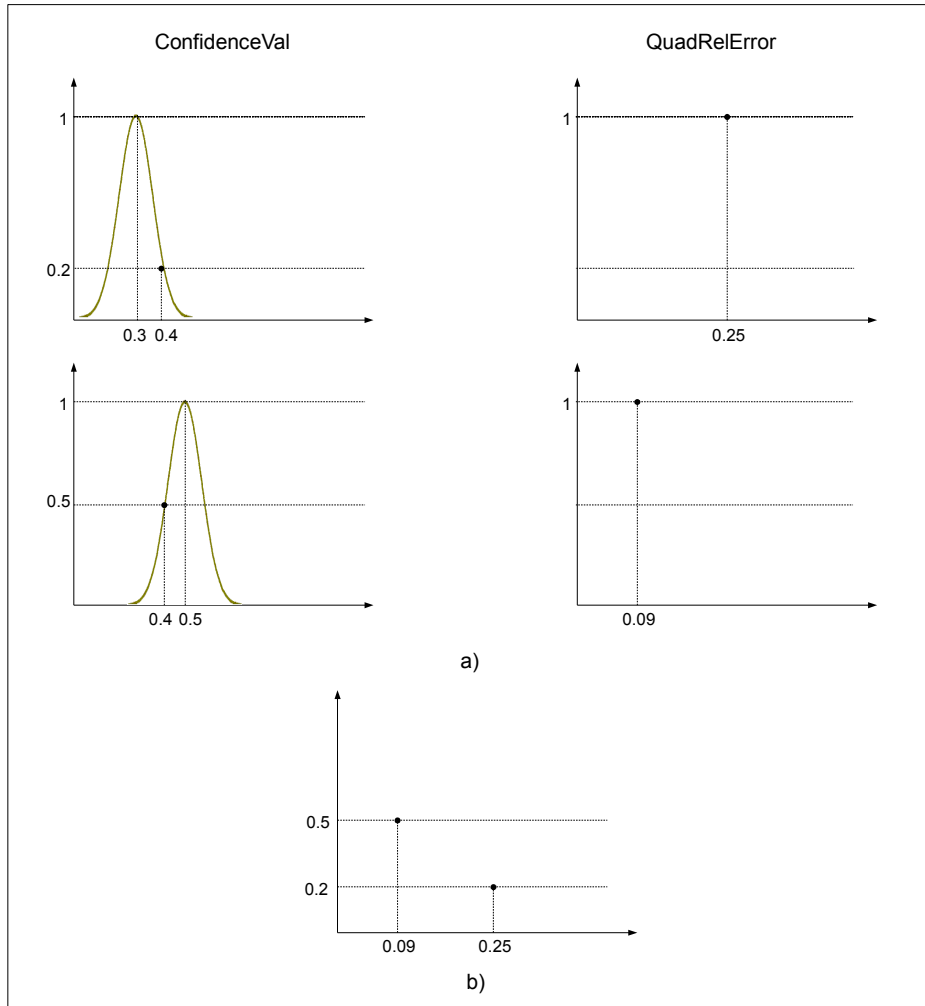
error of the appraised value?. We know that the variance is defined as the expectation of the quadratic error, and also that the mean of the relative error in appraisals is zero by design in the ART testbed. Given the confidence value  $C$ , then, it would be enough for agent  $a$  to gather all the interactions in which agent  $b$  has stated the very same confidence  $C$  in its appraisal and estimate the variance of the relative error in the appraisal given the confidence value  $C$  as the mean of the squares of the errors made. Unfortunately, confidence values will be, in general, scattered along a big range of values, so we can hardly expect to have enough rules with the very same confidence  $C$  to make the estimation accurate. Agent  $a$  can, nevertheless, look at  $C$  as a fuzzy number and use the rules with a confidence value "close enough" to  $C$  in order to improve the estimation. This corresponds to use the taxation vector to compute the output of the fuzzy system (the corrective module) in the following way (see Figure 3.5):

$$\sigma_C^2 = \frac{\sum_{i=1}^{|R|} \mu_i(C) \cdot R[i].QuadRelError}{\sum_{i=1}^{|R|} \mu_i(C)} \quad (3.30)$$

where (see Section 3.5)  $R[i].QuadRelError$  is the relative error made by the appraisal-seller agent in the interaction corresponding to fuzzy rule  $R[i]$  and  $\mu_i(C)$  is the degree to which the value  $C$  belongs to the fuzzy number  $f$  with center  $R[i].ConfidenceVal$ , which we define as:

$$\mu_i(C) = \exp\left(-\frac{(C - R[i].ConfidenceVal)^2}{\alpha^2}\right) \quad (3.31)$$

where the parameter  $\alpha$  determines the width of the fuzzy number (and, thus, the strength with which will fire the rules in the rule base). A value  $\alpha = 0.15$  has proved itself adequate.



**Figure 3.5:** Example of the computation of the output of a FCF for the ART Testbed. *a)* The fuzzy sets representing the two rules of the FCF. The input variable's value is 0.4. *b)* The fuzzy set resulting from combining the output of the two rules. The final value for the filtered variable (the estimated variance of the agent appraisal, in this case) is computed using Equation 3.30 as 0.28.

Figure 3.5 presents one example of output computation for a FCF with only two rules. In the upper part of the figure are the fuzzy sets representing the rules **If** *ConfidenceVal* = 0.3 **then** *QuadRelError* = 0.25 and **If** *ConfidenceVal* = 0.5 **then** *QuadRelError* = 0.09. The value of the input variable (the confidence stated for one appraisal agent *b*) is 0.4. It can be seen the two rules “firing” with different strengths. In the lower part of the figure is represented the fuzzy set resulting from combining the output of the rules. The output of the FCF, that is, the expected value for the variance of the relative error in *b*’s appraisal is computed as (See Equation 3.30) as

$$\sigma_C^2 = \frac{0.09 \cdot 0.5 + 0.25 \cdot 0.2}{0.09 + 0.25} = 0.28$$

A number of refinements can be done to Equations 3.30 and 3.31. For example, we can limit the number of rules that “fire” in the fuzzy system by taking into account in the summatory in Equation 3.30 only those for which  $\mu_i(C)$  is above some given threshold. Another possibility is to dynamically shrink, as simulation goes on and the rule bases get populated, the width of the fuzzy numbers defined in Equation 3.31. Finally, in order to deal with the dynamical nature of the testbed and the fact that agents’ statements can change of meaning with time, we can further modify Equation 3.30 by including the term  $R[i].TimeStep$  which represents the iteration in which the interaction corresponding to rule  $R[i]$  happened:

$$\sigma_C^2 = \frac{\sum_{i=1}^{|R|} \mu_i(C) \cdot R[i].TimeStep^K \cdot R[i].QuadRelError}{\sum_{i=1}^{|R|} \mu_i(C) \cdot R[i].TimeStep^K} \quad (3.32)$$

where we can use the real parameter  $K$  to vary the relative influence of the rules in the computed result, giving more or less importance to more recent interactions.

### 3.8.1 The Reliability Calculation Module

The special features of the ART testbed make difficult the use of several of the quality criteria introduced in Chapter 2<sup>3</sup>. The current implementation of the reliability calculation module for the FCFs, thus, will take into account just the completeness of the rule bases for a given value of the input variable (very roughly speaking, the number of rules that fire in the calculation of the variance). The SPARTAN agent uses the reliability value in two ways:

---

<sup>3</sup> In particular, we can expect to find a big deal of inconsistency between rules.

- To decide whether to buy one appraisal from a given agent  $a$ . If the reliability is high and the expected variance is also high, SPARTAN will tend not to buy the appraisal.
- To decide whether to use one appraisal bought to a given agent  $a$ . Depending on the reliability value Spartan can decide to make use of the appraisal to produce its own final appraisal or not. If the reliability is low, SPARTAN will tend not to use  $a$ 's appraisal. That said, even if the appraisal is not utilized, SPARTAN can use it to refine its model of agent  $a$ , increasing this way the reliability of the corresponding FCF. This makes the behavior of SPARTAN quite conservative (in the sense of not being trustful) during the firsts rounds of the simulation, while trying to build reliable models of the other agents.

### 3.9 Deciding How Much Spend in Creating each Appraisal

Let's suppose that we have a painting to appraise, that we have asked for confidence values to the other agents and that we have estimated for each of them, the expected value of the relative error in their taxations. Now we face two problems:

- How much SPARTAN must spend in the creation of its own opinion? (that is, what value to give to the parameter  $c_g$  in equation 3.14).
- Whose agents SPARTAN must buy its opinions to in order to attain a minimum cost-maximum quality appraisal?

A simple method could be to spend a fixed quantity in the refinement of SPARTAN opinion and, at the same time, buy the appraisal from the  $i$  agents with lesser values for its expected relative error variance. This method is obviously far from optimal. Why should we spend money in worthless opinions?. Don't we have to take into account factors like the accuracy of our own opinion, the accuracy of other agents' opinions or the remaining number of iterations?. On the other hand, this method is too rigid. We surely would like, for example, to spend more money in the first steps of the simulation in order to create as soon as possible an ample base of customers, despite the possible decrease in profits. In the next sections we present two simple and sensible (albeit sub-optimal) ways to tackle this.

#### 3.9.1 Finding a Good Value for $c_g$

Let's suppose that no other agents were available to buy appraisals from. How could we estimate, given a set of paintings to appraise, the optimal

amount of money SPARTAN has to spend in each of the appraisals (that is, the  $c_g$  value in equation 3.14) in order to maximize profit?.

It is an important remark to do before going on with the exposition of the optimization method. The ART testbed uses standard deviations (in the appraisal generation process, see equation 3.14) and mean absolute deviations (in the relative error computation of equation 3.3 and thus in the client share assignment process of equations 3.11 and 3.12).<sup>4</sup> We will need to find a relationship between them. We can do it realizing the fact that if  $X$  is a normal aleatory variable with mean zero and standard deviation  $\sigma$ , then  $|X|$  follows a Half-normal distribution with mean  $\sqrt{2/\pi} \cdot \sigma$  (a proof of this is given in Appendix C). In our case, if the relative error in a taxation is given by a normal distribution with mean zero and standard deviation  $\bar{s}$  (see equation 3.14) then the absolute value of the relative error of the appraisal follows a Half-normal distribution of mean

$$\bar{m} = \sqrt{\frac{2}{\pi}} \cdot \bar{s} \quad (3.33)$$

This means that we can express the expected value of the sum of the absolute values of the relative errors of a set  $T$  of appraisals as a summatory in the form:

$$E(\bar{m}(T)) = \sqrt{\frac{2}{\pi}} \cdot \sum_{q \in T} \bar{s}_q = \sqrt{\frac{2}{\pi}} \cdot \sum_{q \in T} \left( s_q^* + \frac{\alpha}{c_g^q} \right) \quad (3.34)$$

where  $\bar{s}_q$  is the standard deviation of the relative error for appraisal  $q$ ,  $s_q^*$  is the inherent standard deviation assigned by the system to the agent for the era the painting to which appraisal  $q$  refers belongs and  $c_g^q$  is the amount of money spent by the agent in the appraisal.

Take now a look at equation 3.3. The error value  $\epsilon_a$  for the set of appraisals of agent  $a$  is a weighted sum of the absolute values of the relative errors of the set  $T_a$  of appraisals, so its expected value will be:

$$\begin{aligned} E(\epsilon_a) &= E \left( \frac{1}{|T_a|} \cdot \sum_{q \in T_a} \frac{|q.AppVal - q.TrueVal|}{q.TrueVal} \right) \\ &= \frac{\sqrt{2/\pi}}{|T_a|} \cdot \sum_{q \in T_a} \left( s_q^* + \frac{\alpha}{c_g^q} \right) \end{aligned} \quad (3.35)$$

---

<sup>4</sup> Both of them are currently widely accepted and used as a measure of data dispersion [36], however, in our opinion, this blend of measures hardens agent modeling.

Now, we see that there is only one set of magnitudes we can play with in order to minimize the error expected value  $E(\epsilon_a)$ , and they are the amount of money spent in each appraisal. Developing a little further equation 3.35 we have:

$$\begin{aligned} E(\epsilon_a) &= \frac{\sqrt{2/\pi}}{|T_a|} \cdot \left( \sum_{q \in T_a} s_q^* + \sum_{q \in T_a} \frac{\alpha}{c_g^q} \right) \\ &= \frac{\sqrt{2/\pi}}{|T_a|} \cdot \sum_{q \in T_a} s_q^* + \frac{\alpha \cdot \sqrt{2/\pi}}{|T_a|} \cdot \sum_{q \in T_a} \frac{1}{c_g^q} \end{aligned} \quad (3.36)$$

Clearly, the only way to minimize  $E(\epsilon_a)$  is by minimizing the second summatory in Equation 3.36. On the other hand, it is a well known result that, fixed a quantity  $X$ , the minimum for the expression  $\sum_{i=1}^N 1/x_i$  subject to the restrictions  $x_i \geq 0 \forall i$ ,  $\sum_{i=1}^N x_i = X$  is reached for  $x_i = X/N \forall i$ . This leads, in our case, to an interesting result.<sup>5</sup>

**Result 3.1** *When an agent  $a$  acts on its own (that is, when it does not buy appraisals from other agents), whatever be the total amount of money  $M$  it decides to spend in a series of appraisals, the minimum expected value for the error is reached when sharing  $M$  equally between all the appraisals.*

Let's name this common value  $\Phi$ . Substituting in Equation 3.35 and developing the expression a little further, we arrive at a nice and handy equality:

$$\begin{aligned} E(\epsilon_a) &= \frac{\sqrt{2/\pi}}{|T|} \cdot \sum_{q \in T} \left( s_q^* + \frac{\alpha}{\Phi} \right) \\ &= \sqrt{2/\pi} \cdot \left( \frac{\sum_{q \in T} s_q^*}{|T|} + \frac{\alpha}{\Phi} \right) \\ &= \sqrt{2/\pi} \cdot \left( S + \frac{\alpha}{\Phi} \right) \end{aligned} \quad (3.37)$$

where we introduce the value  $S$  as the mean value of the inherent standard deviations  $s_q^*$  associated to the eras corresponding to the whole set of paintings  $T$ . We have then a second result:

**Result 3.2** *When an agent  $a$  acts on its own (that is, when it does not buy appraisals from other agents) and spends a fixed amount of money  $\Phi$  in*

---

<sup>5</sup>Which seems a bit counterintuitive to us. At a first glance one would expect to be more sensible to spend more money in appraisals corresponding to eras for which we have good expertise that in those corresponding to eras we haven't.

each one of a series of appraisals, the expected value of the error  $\epsilon_a$  is  $\sqrt{2/\pi}$  times the sum of  $\alpha/\Phi$  plus the mean inherent standard deviation for the set of paintings to appraise.

We can at this point reformulate our main problem: we want to find, given a series of paintings to be appraised, the optimal common amount  $\Phi$  to be spent in each appraisal in order to maximize the profit in the present iteration.

As everybody knows, profit is incomings minus expenses. The expenses are easy to calculate: if the agent has  $|T|$  appraisals to do and it spends a fixed quantity  $\Phi$  in every appraisal, the total expense will be  $|T| \cdot \Phi$ . Incomes are a little bit trickier, but supposing that agent  $a$ 's appraisals allow it to achieve a preliminary client share of  $\bar{r}_a$  for the next iteration, and knowing that every customer will have to pay an appraisal fee  $F$  they will be equal to  $F \cdot Q \cdot \bar{r}_a$ , where

$$Q = (1 - q)(1 + q + q^2 + \dots + q^{rem}) = (1 - q) \cdot \sum_{i=0}^{rem} q^i \quad (3.38)$$

being  $rem$  the number of remaining iterations, is defined (from Equation 3.7) to reflect the fact that clients are persistent across consecutive iterations.

On the other hand, we have from Equation 3.12 the following expected value for next iteration agent  $a$ 's client share:

$$E(\bar{r}_a) = \frac{E(|C|)}{|A| - 1} \cdot \frac{E(\Sigma)}{E(\epsilon_a) + E(\Sigma)} \quad (3.39)$$

where  $C$  is the set of all the paintings to be appraised in the next iteration and  $A$  is the set of appraiser agents.

Putting it all together and substituting  $E(\epsilon_a)$  for the expression found in Equation 3.37 we have the following expression for  $E(P(\Phi))$ , the expected value of the profit for agent  $a$  in the present iteration given the fixed amount  $\Phi$  of money to be spent in the refinement of each of its appraisals and if no appraisals are bought to other agents:

$$E(P(\Phi)) = Q \cdot F \cdot \frac{E(|C|)}{|A| - 1} \cdot \frac{E(\Sigma)}{E(\Sigma) + \sqrt{2/\pi} \left(S + \frac{\alpha}{\Phi}\right)} - \Phi \cdot |T| \quad (3.40)$$

We can now find the derivative with respect to  $\Phi$  and equal it to zero to obtain the actual value that maximizes  $a$ 's earnings for the present iteration, that is, the amount of money that SPARTAN will spend in each of its own appraisals for every painting in the current iteration.

$$\Phi = \frac{\sqrt{\frac{R \cdot Q \cdot F \cdot E(|C|) \cdot E(\Sigma) \cdot \alpha}{(|A| - 1) \cdot |T|}} - R \cdot \alpha}{E(\Sigma) + S \cdot R} \quad (3.41)$$

where, to summarize:

- $R = \sqrt{2/\pi}$ .
- $Q$  is as defined in Equation 3.38.
- $F$  is the fee clients pay for appraisals.
- $C$  is the total set of paintings to be appraised in the next iteration.
- $E(\Sigma)$  is the expected value of the sum of the mean relative errors that will be made by the agents other than  $a$  during the present iteration.
- $\alpha$  is the constant simulation parameter introduced in Equation 3.1.
- $A$  is the set of agents.
- $T$  is the set of paintings to be appraised by agent  $a$  during the present iteration.
- $S = \sum_{q \in T} s_q^*$  is the arithmetic mean of the value of  $a$ 's inherent standard deviations associated to the eras corresponding to the set of paintings  $T$ .

There are two terms in the previous equation whose values are unknown to the agent in the moment of the optimal appraisal calculation,  $Q$  and  $E(\Sigma)$ <sup>6</sup>. It is little the agent can do to know or estimate  $Q$ , other than guessing. Designers of the ART Testbed decided the number of iterations for each simulation to be random in prevention of exploitative behaviors in the final part of the simulations<sup>7</sup>. So no agent can have this information, even if it would use it only for its own guidance during the appraising process. This is quite inconvenient, given that it is easily shown that  $Q$ 's value diminishes as iterations go on, from  $Q \approx 1$  in the first iterations to  $Q = (1 - q)$  in the last one. This can considerably influence the result of the computation of  $\Phi$ . However, examining equation 3.38, we realize that, for a large enough number of iterations in the simulation, the percentage of those for which  $Q$ 's value is substantively different from one is small enough to allow the agent to disregard the issue (considering  $Q$  to be equal to the unity all along the

---

<sup>6</sup>In the current implementation of the simulator  $|C|$  is constant for every iteration and, consequently, so is  $E(|C|)$  and there is no problem. In any case we assume that  $E(|C|)$  is known

<sup>7</sup>And even not to make public the probability distribution it follows



simulation) without incurring in too much penalties. That said, probably some good guessing heuristic could give better results.

The situation regarding  $\Sigma$  is a bit better. The agent can not know, of course, what total mean error will make the other agents during the current iteration. It can, however, calculate what the value of this error was in the last iteration using Equations 3.7 and 3.12 and use this value as an estimator for the total mean error in the current iteration. From Equation 3.7 we can express agent  $a$ 's preliminary client share for the present iteration as:

$$\bar{r}_a = \frac{r_a - q \cdot r'_a}{(1 - q)} \quad (3.42)$$

where  $r_a$  and  $r'_a$  are the final client shares for the current iteration and the previous one, respectively. We can then use Equation 3.12 to obtain, after simplifying, the following value for  $\Sigma'$ , the total mean error committed by the rest of the agents in the previous iteration:

$$\Sigma' = \frac{\epsilon'_a \cdot \bar{r}_a \cdot (|A| - 1)}{|C| - \bar{r}_a \cdot (|A| - 1)} \quad (3.43)$$

where  $\epsilon'_a$  is the mean relative error made by agent  $a$  during the previous iteration. Finally, we can estimate  $E(\Sigma)$  as:

$$E(\Sigma) = \lambda \cdot \Sigma' \quad (3.44)$$

where  $0 \leq \lambda \leq 1$ .  $\lambda$  is a learning parameter intended to model the fact that total error will decrease as simulation runs. It is interesting to remark that this estimator will perform substantially better in scenarios with a big number of agents where fluctuations in individual errors will tend to cancel each other <sup>8</sup>.

### 3.9.2 Deciding Which Appraisals to Buy

Once determined  $\Phi$ , the task of deciding which appraisals to buy for each painting can be overtaken. The method will be quite straightforward. Once the set of confidence values for one painting has been received from the set of appraisal agents, we will use the FCF introduced in Section 3.8 in order to determine a vector of expected variances for each appraiser agent's appraisal.

---

<sup>8</sup>Another unexplored and straightforward possibility is to heuristically estimate the expected value of  $\Sigma$  as

$$E(\Sigma) = \alpha \cdot \epsilon'_a$$

where we could make  $\alpha > 0$  depend on the difference between  $r_a$  and  $r'_a$

Then we will arrange this vector in order of increasing expected variances and, finally, we will start picking appraisals from the beginning of the list, one by one, as long as the expected value of final profit increases, stopping at the point the expected value of the final profit starts to decline. Let us see how this is done.

For the sake of simplicity, let's study, first, the case in which one agent  $a$  is at the beginning of an iteration. Suppose the agent has asked for appraisal offers for the first painting  $p$  and suppose that  $b$  is the agent that can provide the appraisal with lower expected variance. Let  $A_a$  be  $a$ 's own appraisal and  $A_b$  be the appraisal  $b$  can provide. Let  $Var(A_a)$  denote the variance of  $A_a$  and  $Var'(A_b)$  denote the expected value of the variance of  $A_b$  computed by the corresponding FCF. How to determine the worthiness of buying  $A_b$ ? If agent  $a$  buys  $A_b$  the expected value of the mean error made by agent  $a$  during the present iteration,  $E(\epsilon_a)$  will decrease and this will lead to a bigger client share expectation in the next iteration and, thus, greater expected profit. On the other hand, the agent has to spend money in the buying of  $A_b$ , and this means a decrease in the final earnings. Summarizing, in order to decide whether to buy  $A_b$  or not<sup>9</sup> we have to answer, first, the following questions:

- How the expected value of the mean error made by agent  $a$  in the present iteration,  $E(\epsilon_a)$ , will decrease by buying  $b$ 's appraisal?
- What impact will this diminution have in the final earnings of the agent, taking into account the cost of buying  $A_b$ ?

Let us first answer the first question. Once known  $\Phi$ , the optimum amount to be spent by the agent in each appraisal, we can calculate from equation 3.35 the expectation of the mean relative error which will be made by the agent in the current iteration, provided that no appraisal is bought from any other agent as:

$$E(\epsilon_a) = \frac{\sqrt{2/\pi}}{|T_a|} \cdot \sum_{q \in T_a} \left( s_q^* + \frac{\alpha}{\Phi} \right) = \sqrt{2/\pi} \cdot \left( S + \frac{\alpha}{\Phi} \right) \quad (3.45)$$

where  $S = 1/|T_a| \cdot \sum_{q \in T_a} s_q^*$  is the mean value of the base expertise of agent  $a$  for each of the eras to which the paintings in  $T_a$  belong.

How will diminish  $E(\epsilon_a)$  with the buying of  $A_b$ ? As we can see in the second term of Equation 3.45, the contribution of painting  $p$  to  $E(\epsilon_a)$  can be found in the summatory as.

---

<sup>9</sup>It is important to remark that, given that  $A_b$  is the appraisal with lesser expected variance, if it is not worthwhile for agent  $a$  to purchase  $A_b$ , neither will be none of the remaining ones

$$\sigma_p = \left( s_p^* + \frac{\alpha}{\Phi} \right) \quad (3.46)$$

That is,  $\sigma_p$  (see Equation 3.18) is the standard deviation of the relative error in  $A_a$ ,  $a$ 's appraisal for painting  $p$ . So, if we purchase  $b$ 's appraisal,  $A_b$  and combine  $A_a$  and  $A_b$ , we must replace  $\sigma_p$  in Equation 3.45 by the value of the standard deviation of the relative error of the combined appraisal. This value can be computed using Equation 3.25 in Section 3.7 as follows:

$$\sigma_{ab} = \sqrt{\frac{Var(A_a) \cdot Var'(A_b)}{Var(A_a) + Var'(A_b)}} \quad (3.47)$$

where  $Var(A_a) = \sigma_p^2$  and  $Var'(A_b)$ , an estimator of  $Var(A_b)$ , is the expected variance for appraisal  $A_b$ , computed by the corresponding FCF from the confidence value stated by agent  $b$  respecting its appraisal for painting  $p$ .

So, the new expected mean relative error for  $a$ 's set of appraisals will be:

$$\begin{aligned} E'(\epsilon_a) &= \frac{\sqrt{2/\pi}}{|T_a|} \cdot \sum_{q \in T_a, q \neq p} \left( s_q^* + \frac{\alpha}{\Phi} \right) + \frac{\sqrt{2/\pi}}{|T_a|} \cdot \sigma_{ab} \\ &= \frac{\sqrt{2/\pi}}{|T_a|} \left( \sigma_{ab} + \sum_{q \in T_a, q \neq p} \left( s_q^* + \frac{\alpha}{\Phi} \right) \right) \end{aligned} \quad (3.48)$$

and we can now express the expected decrease in  $a$ 's mean relative error in the present iteration caused by the buy of  $A_b$  as:

$$E(\Delta\epsilon_a) = E(\epsilon_a) - E'(\epsilon_a) = \frac{\sqrt{2/\pi}}{|T_a|} \cdot \left( s_p^* + \frac{\alpha}{\Phi} - \sigma_{ab} \right) \quad (3.49)$$

where  $s_p^*$  is the base expertise of agent  $a$  for the era to which painting  $p$  belongs.

We can now face the second problem, once known what the mean relative error would be if the agent bought  $A_b$ , a criterion must be found to help the agent to decide whether actually buy  $A_b$  or not. The criterion will be very simple,  $a$  will buy  $b$ 's appraisal  $A_b$  as long as the expected benefit produced for the decrement in the mean relative error was superior to the expense caused by the buying of  $A_b$ . We have seen how to do something similar in Section 3.9.1. From Equation 3.12 we can express the difference between the expected preliminary client share if agent  $a$  buys  $b$ 's appraisal  $A_b$  and the expected preliminary client share if agent  $a$  does not buy it (that is, the expected value of the increment of  $a$ 's preliminary client share for the next iteration caused by buying  $A_b$ ) as:

$$E(\Delta\bar{r}_a) = \frac{|C|}{|A| - 1} \left( \frac{E(\Sigma)}{E'(\epsilon_a) + E(\Sigma)} - \frac{E(\Sigma)}{E(\epsilon_a) + E(\Sigma)} \right) \quad (3.50)$$

where  $C$  is the total set of paintings to be appraised in the next iteration,  $A$  is the set of appraisers and  $E(\Sigma)$  is the expected value of the sum of the mean relative errors that will be made by the agents other than  $a$  during the present iteration (which we will estimate as explained in Section 3.9.1).

This expected increment in  $a$ 's preliminary client share for the next iteration will produce an expected increment in the profit in the subsequent iterations equal to:

$$E(\Delta G) = F \cdot Q \cdot E(\Delta\bar{r}_a) - c_p \quad (3.51)$$

where  $F$  is the fee paid by costumers to the appraiser agents,  $Q$  is as defined in Equation 3.38 and  $c_p$  is the fee agent  $a$  has to pay to agent  $b$  for its appraisal ( $c_p$  is constant and equal for all agents during the simulation). Thus, the procedure to decide whether to buy  $A_b$  or not is quite straightforward: agent  $a$  computes  $E(\Delta G)$  and buys  $A_b$  only if its value its positive.

And what about buying a third appraisal  $A_c$ ?. Actually the procedure above can be generalized to any number of appraisals. It can be easily proved that buying  $n$  appraisals to  $n$  appraiser agents is equivalent to buying the optimal combination of the  $n$  appraisals (as explained in Section 3.7) to one agent for  $n$  times the fee  $c_p$ . Suppose, for example, that agent  $a$  has to decide whether it is worth to buy appraisals  $A_b$ ,  $A_c$  and  $A_d$  to the agents  $a$ ,  $b$  and  $c$ . The only thing it has to do is compute the variance of the optimal combination of its own appraisal  $A_a$  with  $A_b$ ,  $A_c$  and  $A_d$  using Equation 3.29 and take its square root to obtain the standard deviation of the combined appraisal  $\sigma_{abcd}$ . Then it can use Equation 3.48 to determine the expected error  $E'(\epsilon_a)$  (changing  $\sigma_{ab}$  for  $\sigma_{abcd}$ ) and Equation 3.50 to obtain the expected increment in the preliminary client share. Finally, the expected increase in the profit due to the decrement in the mean relative error will be:

$$E(\Delta G) = F \cdot Q \cdot E(\Delta\bar{r}_a) - 3 \cdot c_p$$

Note the main difference with respect to Equation 3.51:  $c_p$  is multiplied by 3, the number of purchased appraisals. The way SPARTAN will decide the actual set of appraisals to buy is the following: it will pick, in order and one by one, the appraisals with smaller expected variances and will calculate the expected error and expected increase in profit for each of the combinations (for example, first just  $A_a$ , then  $A_a$  and  $A_b$ , then  $A_a$  and  $A_b$  and  $A_c$  and so on) as long as the expected value of final profit increases, and will stop at

the point the expected value of the final profit starts to decline<sup>10</sup>.

Finally, until now, we have only studied the case where painting  $p$  is the first one in the iteration. What about the paintings other than the first?. The answer is simple: after processing the first painting, the expected mean error for the current iteration  $E(\epsilon)$  will decrease in value to  $E'(\epsilon)$ . Every subsequent painting can be processed in the same way than the first one, when it arrives its turn to be appraised, provided that the value of the expected mean error for the current iteration  $E(\epsilon)$  is updated to the value  $E'(\epsilon)$  obtained in the processing of the previous painting. The whole process is summarized in Algorithm 3.2. The algorithm is pretty well auto-explicative. The only new point is the new error variable  $\epsilon''$ , necessary to retrieve, after the processing of each painting, the value of  $\epsilon'$  in the last but one iteration of the loop<sup>11</sup>.

### 3.10 Results

The global behavior of agents in ART testbed experiments is very sensitive to even small changes in the environment or in the particular behavior of single agents. In order to try to overcome this problem, two series of simulations have been carried out, using two sets of agents, a first one (Set A) with several of the best competitors in the 2006 International ART Competition (i.e. IAM, Frost, Neil, and Sabatini), and a second one (Set B) with new agents synthesized to be more trusty. Ten simulations have been done in each series. In five of them our agent, SPARTAN, uses FCFs in order to translate the certainty values provided by the other agents to variances, in the remaining five simulations SPARTAN don't uses FCFs, that is, he assumes the other agents to be trustworthy and to talk the same language than itself. A representative set of results is shown in Figure 3.6.

As a consequence of ART Testbed sensitivity to initial conditions, the inherent random nature of the simulations makes the amount of money earned by the agents in every run to be very variable. Therefore one cannot simply take money as an absolute performance measure. Though other methods

---

<sup>10</sup>This method can be non optimal depending on the values of  $F$  and  $c_p$ . We have adopted it for the sake of simplicity.

<sup>11</sup>The algorithm exits the loop when the combination of the  $N$  best appraisals is worse than the combination of the  $N - 1$  best appraisals. In this point, after buying those  $N - 1$  appraisals and in order to update  $\epsilon$ , the expected mean error for the current iteration, correctly for the processing of the next painting, the algorithm needs to remember what the value of  $\epsilon'$  was in the previous iteration of the loop, when only the  $N - 1$  best appraisals were combined.

---

**Algorithm 3.2** One iteration of SPARTAN in the ART testbed.

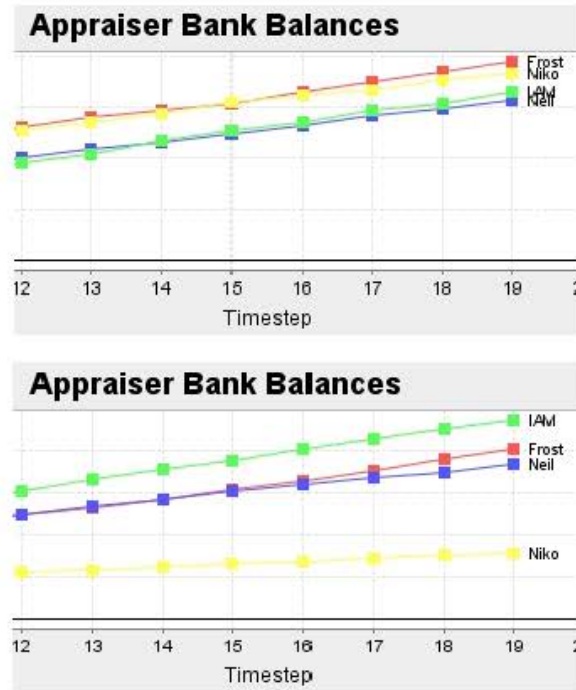
---

```

1: procedure ONEARTITERATION( $T, \Sigma, \alpha, \Phi, \{s^*\}, F, c_p$ )
2:    $\epsilon \leftarrow \frac{\sqrt{2/\pi}}{|T|} \sum_{q \in T} (s_q^* + \frac{\alpha}{\Phi})$ 
3:   for each painting  $p \in T$  do
4:     ask all other agents for appraisals for  $p$ 
5:      $L \leftarrow$  List, in increasing order, of the expected variances output
   by the FCFs of the relative errors in other agents' appraisals
6:      $L \leftarrow (s_p^* + \frac{\alpha}{\Phi})^2 + L$ 
7:      $N \leftarrow 1$ 
8:      $\Delta G \leftarrow 0$ 
9:      $\epsilon', \epsilon'' \leftarrow \epsilon$ 
10:    loop
11:       $N \leftarrow N + 1$ 
12:      if  $N > \text{length}(L)$  then
13:         $\epsilon'' \leftarrow \epsilon'$ 
14:        Break
15:      end if
16:       $\sigma_N \leftarrow \sqrt{2/\pi \cdot \frac{\prod_{i=1}^N L_i}{\sum_{i=1}^N \prod_{j \neq i} L_j}}$ 
17:       $\epsilon'' \leftarrow \epsilon'$ 
18:       $\epsilon' \leftarrow \epsilon - \frac{\sqrt{2/\pi}}{|T|} \cdot (s_p^* + \frac{\alpha}{\Phi} - \sigma_N)$ 
19:       $\Delta \bar{r} \leftarrow \frac{|C|}{|A|-1} \left( \frac{\Sigma}{\epsilon' + \Sigma} - \frac{\Sigma}{\epsilon + \Sigma} \right)$ 
20:       $\Delta G' = F \cdot Q \cdot \Delta \bar{r} - (N - 1) \cdot c_p$ 
21:      if  $\Delta G' > \Delta G$  then
22:         $\Delta G \leftarrow \Delta G'$ 
23:      else
24:        Break
25:      end if
26:    end loop
27:    buy appraisals corresponding to the  $N - 1$  first entries in  $L$ 
28:    compute weights, if necessary, according to equation 3.28
29:    send weights vector to the simulator
30:     $\epsilon \leftarrow \epsilon''$ 
31:  end for
32: end procedure

```

---



**Figure 3.6:** *SPARTAN* (under the nickname *Niko*) versus *IAM*, *Neil* and *Frost*. Bottom: Results without using FCFs. Top: Results using FCFs.

**Table 3.1:** Results of the experiments.

	With FCF		Without FCF		%	$\tau$
	ME	SD	ME	SD		
<b>A</b>	0,903	0,059	0,741	0,033	21,9	5,35
<b>B</b>	0,89	0,096	0,75	0,125	18,7	1,99

may apply (ranking, etc), we want a method able to keep accurately the distance between agents in the different runs, so we have decided to normalize the results by dividing the money earned by *SPARTAN* by the money earned by the remaining best agent. This gives us an adimensional measure of *SPARTAN*'s *efficiency* that helps the comparison.

The results of the experiments can be seen in Table 3.1, where ME stands for Mean Efficiency and SD stands for Standard Deviation. Results are very similar in both series, although slightly better in the case of Set A, the more competitive agents. The improvement in efficiency is about 20% in both series. The t-values for the unequal variances Student's t-Test guarantees the statistical significance of the results with high probability (greater than

0.999 for Set A and greater than 0.95 for Set B).

FCF (along with the other tricks and techniques introduced in this chapter) allowed SPARTAN, to win the 4th position out of 18 participants in the 2nd international ART competition in AAMAS 2007, May 14-18, 2007 in Hawaii. A summary of the competition results and competitor teams can be found in Appendix E. It would be interesting to analyze the differences in the strategies of the SPARTAN agent and the other agents involved in the competition in order to discover the respective strengths and weaknesses. Unfortunately, competing teams are generally not too keen to disclose their tactics, but it seems that the main reason of the relative stronger performance of other agents has to be found in strategies directed to the formation of strong coalitions of cooperating agents. This is an aspect to which SPARTAN did not pay attention enough. Perhaps in the future we will be able to integrate those kind of strategies into it. Although it was not in the original plan, it would be interesting to see if the use of FCF's can give the agent a more competitive edge.

### 3.11 Conclusions and Proposals to the ART Testbed Designers

In this chapter, we have discussed several aspects of the ART Testbed and presented the techniques and algorithms behind the appraiser agent SPARTAN, specially the application of FCFs to the estimation of other agents's appraisals variances. We have also presented the promising results of our approximation to the problem. As a finish, and ever congratulating the ART testbed designers and team in general for a well-done job, we would like to make several proposals that, in our opinion, could improve the design of the testbed and its overall use experience:

1. The simulator lacks a mechanism allowing the agents to be aware of the reputation values other agents are assigning to them. That mechanism often exist in real-life scenarios and could be easily incorporated to the testbed.
2. Five agents are definitely too few to make reputation requests useful. After all, every agent has its own experience which allows it to evaluate as accurately as any other agent the reliability of the other agents without having to ask for it and then translate the answer. Reputation requests would be useful in scenarios with too many agents as to



possibly know all of them <sup>12</sup>.

3. As stated in Section 3.3 the system does not really encourage deceit. The benefits of deceit (that is, selling an appraisal with a big error) are shared equally amongst all the agents (except the deceived agent, of course). Perhaps the capacity to deceive other agents ought to be rewarded.
4. Finally, and perhaps most important, the formula used for the computation of the standard deviation of the relative error in one appraisal:

$$s = \left( s^* + \frac{\alpha}{c_g} \right)$$

allows one agent to produce appraisals with arbitrarily large error expectation (by assigning a near-zero value to  $c_g$ ). One agent fooled to buy one of such appraisals will probably not have any new customer in the next iteration and, worse yet, will cause the rest of agents to share almost evenly the customer pool, independently of its performance in the current iteration. That seems a bit drastic and unfair<sup>13</sup>. In order to fix the problem, a new formula for the standard deviation of the relative error is proposed:

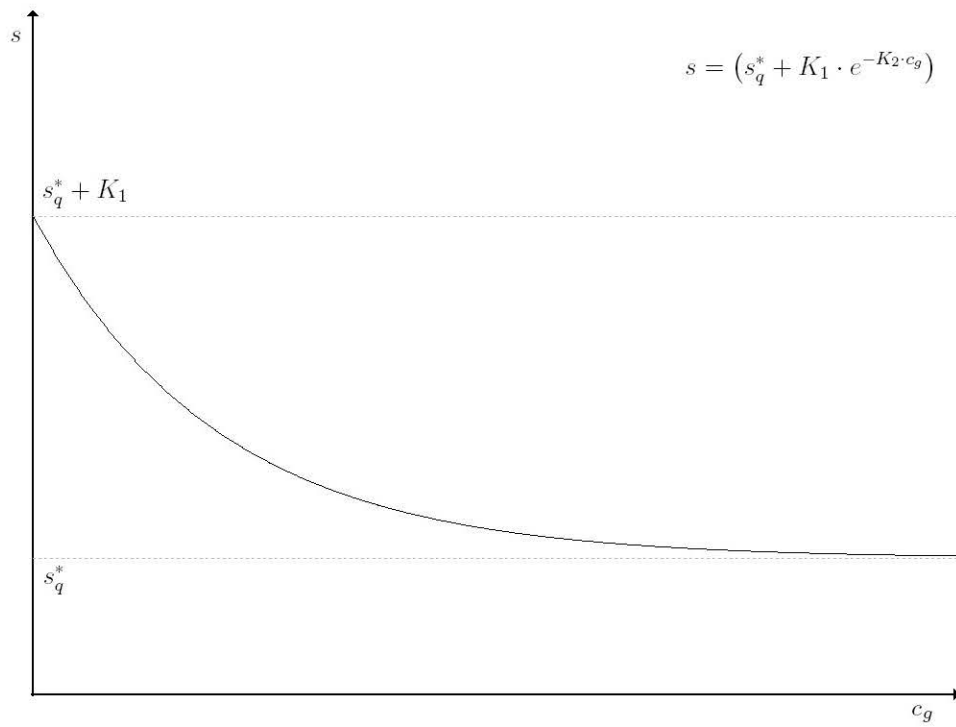
$$s = (s_q^* + K_1 \cdot e^{-K_2 \cdot c_g}) \quad (3.52)$$

This new formula nicely bounds the standard deviation of the relative error  $s$  between  $s^* + K_1$  and  $s^*$ , decreasing when  $c_g$  increases, as desired (See Figure 3.7). The parameter  $K_2$  can be used to adjust the steepness of the curve.

---

<sup>12</sup>This issue has been addressed to in the last competitions, with the new versions of the simulator

<sup>13</sup>Moreover, if it's two or more agents that buy one of these huge variance appraisals, the result will tend to be the same, because, with high probability, one of the errors is bound to be much higher than the others



**Figure 3.7:** A new formula for the computation of the standard deviation of the relative error in appraisals.



## Chapter 4

# From Swarm Intelligence to Crew Intelligence: Bar Systems

### 4.1 Introduction

This chapter presents Bar Systems: a family of very simple algorithms for different classes of complex optimization problems in static and dynamic environments by means of reactive multiagent systems. Bar Systems belong to the family of Crew Intelligence algorithms, an extension of Swarm Intelligence algorithms also introduced in this chapter, that endows individual agents with additional communicative and local planning abilities. Bar Systems are loosely inspired in the behavior that a crew of bartenders can show while serving drinks to a crowd of customers in a bar or pub. We will see how Bar Systems can be applied to CONTS, a NP-hard scheduling problem, and how they achieve much better results than other greedy algorithms in the “nearest neighbor” style. We will also prove this framework to be general enough to be applied to other interesting optimization problems like generalized versions of the flexible Open-shop, Job-shop and Flow-shop problems.

The chapter is organized as follows: in the next section we will make a short introduction to the main Swarm Intelligence concepts and techniques. In section 4.3 we will introduce the novel concept of Crew Intelligence and compare it to Swarm Intelligence. In section 4.4 we will present and formalize the concept of Bar System, in section 4.5 we present the CONTS problem, a NP-hard scheduling problem for multi agent systems which will serve us to test the performance of Bar Systems. In sections 4.6 and 4.7 we will see how to solve the CONTS problem using a Bar System and we will comment the results. Finally, in section 4.8, we will draw some conclusions and we will discuss some directions towards which future work can be directed.

## 4.2 Swarm Intelligence

A commonly accepted and used definition of the term Swarm Intelligence is: “the property of a system whereby the collective behaviors of (unsophisticated) agents interacting locally with their environment cause coherent functional global patterns to emerge”. The origin of the term is to be found in the observation of social insect colonies and its paradigm is an ant colony. In it, individual ants’ behavior is controlled by a small set of very simple rules, but their interactions (also very simple) with the environment allow them to solve complex problems (such as finding the shortest path from one point to another one). Ant colonies (and the same could be said about human beings) are intelligent systems with great problem solving capabilities, formed by a quantity of relatively independent and very simple subsystems which do not show individual intelligence. It is the “many nitwits make a witty one” phenomenon of emergent intelligence.

A bunch of Swarm Intelligence-inspired problem solving techniques have appeared over the last few years. Three of the most successful such techniques currently in use are Ant Colony Optimization [24], Particle Swarm Optimization [47] and Stochastic Diffusion Search [6]. Ant Colony Optimization techniques, also known as Ant Systems, are based in ants’ foraging behavior, and have been applied to problems ranging from determination of minimal paths in TSP-like problems to network traffic rerouting in busy telecommunications systems. Particle Swarm Optimization techniques, inspired in the way a flock of birds or a school of fish moves, are general global minimization techniques which deal with problems in which a best solution can be represented as a point or surface in an n-dimensional space. Stochastic Diffusion Search is another generic population-based search method in which agents perform cheap, partial evaluations of a hypothesis (a candidate solution to the search problem) and then share information about hypotheses (diffusion of information) through direct one-to-one communication. As a result of the diffusion mechanism, high-quality solutions can be identified from clusters of agents with the same hypothesis.

Swarm Intelligence techniques present several advantages over more traditional ones. On one hand, they are cheap, simple and robust; on the other hand, they provide a basis with which it is possible to explore collective (or distributed) problem solving without centralized control or the provision of a global model. Over the last years they have found application in a wide variety of domains: collective robotics, vehicle navigation, planetary mapping, streamlining of assembly lines in factories, coordinated robotic transport, banking data analysis and much more. The interested reader can find a lot of useful references about self-organization and Swarm Intelligence theory and applications in [4, 8, 7, 28, 43, 50].

### 4.3 Swarm Intelligence vs. Crew Intelligence

No doubt, Swarm Intelligence techniques have proved its usefulness over the last years. Nevertheless, in our opinion, their applicability and effectiveness is somewhat limited by the simplicity of the individual agents in the swarm. In the typical Ant Colony Optimization systems, for example, ants behavior is purely reactive and communication between ants is only allowed through the environment, in the form of a pheromone trail. One can't help but wonder whether it would be possible to increase the individual communication and problem solving capabilities of the agents in a Swarm Intelligence system, while at the same time maintaining the desirable features of cheapness, locality, decentralization, simplicity and robustness and what impact would it have in the overall behavior of the system.

It turns out to be that it is possible to find such systems in the real world, specially in those situations where people have to coordinate themselves in a highly dynamic environment in order to solve some kind of scheduling process. Examples are a vessel crew, a staff of bartenders serving pints in a pub or a soccer team. These kind of systems are characterized by highly dynamic environments where tasks of different classes quickly appear and disappear and have to be carried out in a timely fashion. Coordination between people in this kind of systems is very important, but is not attained, typically, by means of some centralized global procedure. The behavior of the individual agents is mainly reactive (they react to the appearance and disappearance of tasks) but, at the same time, and this differentiate them from classical Swarm Intelligence systems, they make a limited<sup>1</sup> use of their "human" abilities (complex communication, reasoning, local planning ...) to coordinate themselves and increase the problem solving effectiveness of the system. We have chosen to christenize those kind of empowered Swarm Intelligence Systems with the name of Crew Intelligence Systems<sup>2</sup>.

The class of systems we present in this chapter, Bar Systems [20, 21], belonging to the broader class of Crew Intelligence systems, are reactive multi agent systems whose behavior is loosely inspired in that of a crew of bartenders. Three traits distinguish them:

- They are well suited for finding approximate solutions for large and complex scheduling real-time problems in highly dynamic environments.

---

<sup>1</sup>Limited mainly by time constraints.

<sup>2</sup>Crew: a group of people associated together in a common activity or by common traits or interests. (From the *Merriam-Webster Online Dictionary* [45])

- The behaviors of individual agents are directed towards the maximization of a local affinity function. This individual behavior results in the whole system tending to the minimization of a global cost function.
- Individual agents are endowed with more or less complex communication and local planning abilities which increase the problem solving capabilities of the system.

## 4.4 Bar Systems

Anybody who has tried to get served a pint in a bar crowded with customers will have had more than enough time to wonder with boredom about the method used by waiters, if there is any, to decide which customer to pay attention to at each time. Sometimes there is not much point, to be served before, in having been waiting for a long time or in yelling at the waiter. Details like the bar area where the customer is, his/her sex, whether the waiter knows him/her or whether the waiter likes the customer's face determine to a high extent the way in which orders are served.

Let us examine the situation from the bartenders' point of view: a heap of customers are ordering drinks at once, new ones arrive all the time, and the bartenders have to do all their best to serve them. Of course, they cannot do it in an random way; they have to try to maximize some kind of utility function which will typically take into account aspects such as average serving time, average serving cost or average customer (and boss) satisfaction. They will have to pay attention, then, to facts such as that some of them can prepare certain drinks more quickly or better than others, that the order in which the drinks are served influences the time or the total cost of serving them, and that also moving from one place in the bar to another costs time. All of this without forgetting, on one hand, that the order in which orders were taken has to be respected as much as possible and, on the other hand, that they have to try to favor the best customers by giving them special preferential attention and best service and keeping them waiting for a shorter time.

The problem is not at all trivial, (actually we will see that it can be proved to be NP-hard), bartenders have to act in a highly dynamic, asynchronous and time-critical environment, and no obvious greedy strategy (such as serving first the best customer, serving first the nearest customer or serving first the customer who has arrived first) gives good results. Nevertheless, a staff of good bartenders usually can manage to serve a lot of customers in such a way that the vast majority of them were, more or less, satisfied. The way they accomplish the task seems to have little to do with any global planning or explicit coordination mechanisms but, arguably, with trying to maximize,

every time they choose a customer to serve, some local utility function which takes into account aspects like the importance of the customer, the cost for the waiter of serving her/him and the time that he/she has been waiting for service. On the other hand, nevertheless, waiters use to communicate with each other in order to do their job more efficiently (asking other waiters for something or deciding which waiter serves to each customer, for example), and it is undeniable that they employ often some kind of local planning method (even if it consists simply in serving two customers at once when they ask for the same or a similar beverage).

In the next section, we will try to give a general formalization of this type of problem solving behaviors, which we call Bar Systems.

#### 4.4.1 Definition

We will define a Bar System as a quintuple  $(E, A, T, F, C)$  where:

1.  $E$  is a (physical or virtual) environment. The state of the environment at each moment is determined by a set of state variables  $V_E$ . One of those variables is usually the time. We define  $S$  as the set of all possible states of the environment  $E$ , that is, the set of all the possible simultaneous instantiations of the set of state variables  $V_E$ .
2.  $A = \{a_1, a_2, \dots, a_N\}$  is a set of agents situated into the environment  $E$ . Each agent  $a_i$  can have different problem-dependent properties (i.e. weight, speed, location, response time, maximum load...).
3.  $T = \{t_1, t_2, \dots, t_M\}$  is a set of tasks to be accomplished by the agents within the environment  $E$ . Each task has its own conditions of appearance and disappearance which are not known by the agents. Each task  $t_i$  has associated:
  - $imp(t_i)$ . A real function of  $S$  that reflects the importance of the task  $t_i$  in the current state of the environment.
  - $urg(t_i)$ . A positive real function of  $S$  which represents the urgency of task  $t_i$  in the current state of the environment. It will be usually a nondecreasing function of time.

Additionally, for each agent  $a_i$  and each task  $t_j$ , we have:

- $pre(a_i, t_j)$ . A set of preconditions over  $V_E$  which determine whether the task  $t_j$  can be done by agent  $a_i$ .
- $post(a_i, t_j)$ . A set of postconditions over  $V_E$  which describes the resulting state of the environment after task  $t_j$  has been done by agent  $a_i$ .



- $cost(a_i, t_j)$  A function of  $V_E$  that reflects the cost for agent  $a_i$  to execute the task  $t_j$  in the current state of the environment. We will come back to this function in section 4.4.1.1.
4.  $F : S \times A \times T \rightarrow \Re$  is the function which reflects the degree to which agents are "attracted" by tasks. Given a state  $s$  of the environment, an agent  $a_i$  and a task  $t_j$   $F(s, a_i, t_j)$  must be defined in a way such that it increases with  $imp(t_j)$  and  $urg(t_j)$  and it decreases with  $cost(a_i, t_j)$ .
  5.  $C$  is the global cost function to be minimized by the Bar System.  $C$  can have different forms. It can be, for example, the total time needed to finish all the tasks, the mean time, the total cost, etc.

#### 4.4.1.1 The $cost(a_i, t_j)$ function

As stated in the preceding section the function  $cost(a_i, t_j)$  reflects the cost for agent  $a_i$  to execute the task  $t_j$  in the current state of the environment. This cost can be divided in three parts:

$$cost(a_i, t_j) = cost_{pre}(a_i, t_j) + cost_{task}(a_i, t_j) + cost_{post}(a_i, t_j) \quad (4.1)$$

where:

- $cost_{pre}(a_i, t_j)$  reflects the cost for  $a_i$  to make the environment fulfill the preconditions of task  $t_j$  (this can include the cost of stop doing his current task). If agent  $a_i$  is unable to adapt the environment to the preconditions of task  $t_j$  then we can define it as infinite.
- $cost_{task}(a_i, t_j)$  is the cost for  $a_i$  to actually execute  $t_j$  once preconditions are fulfilled and  $cost_{post}(a_i, t_j)$ . If agent  $a_i$  is unable to carry out task  $t_j$  by itself then we can define it as infinite.
- $cost_{post}(a_i, t_j)$  has to reflect the fact that different agents modify the environment differently as a result of the actions they undertake in order to carry out the same task, thus hindering or facilitating to a greater or lesser extent the execution of other pending tasks (that is, the resulting state of the environment after an agent  $a_i$  has performed some task  $t_j$ , given by  $post(a_i, t_j)$ , can modify  $cost_{pre}(a_p, t_q)$  for one or more agents  $a_p$  and tasks  $t_q$ ). Think, for example, about two bartenders. Both of them can prepare a given cocktail, say a Dry Martini, but one of them makes use of the cocktail shaker to do it, while the other one does not. Which one of the bartenders prepares the Martini has great influence over the cost of preparing, say, a Mojito at a later time. Mojito preparation requires a cocktail shaker that will have to

be cleaned if it has been used to prepare the Dry Martini. A suitable definition of  $cost_{post}(a_i, t_j)$  would be:

$$cost_{post}(a_i, t_j) = K \cdot \sum_{t \in T, t \neq t_j} d(s, pre(t)) - d(s', pre(t)) \quad (4.2)$$

for a suitable value of the parameter  $K$ , where  $s$  and  $s'$  are, respectively, the state of the environment before and after agent  $a_i$  performs task  $t_j$  and  $d(a, b)$  is some distance that estimates the cost of making the environment change from state  $a$  to state  $b$ .

#### 4.4.2 Specialization in Bar Systems

The cost of carrying out a given task is variable over the set of agents, that is, different agents will incur in different costs carrying out the same task. Agents, then, will tend to prefer those tasks that are easier for them (have a lesser cost), which will have a greater value for the affinity function  $F$ . This provides for specialization. Nevertheless, if no specialist is available for a given task, another less suitable agent will eventually carry it out at the moment the urgency of the task causes the value of the affinity function rise enough.

#### 4.4.3 Bar System's Dynamics

In Bar Systems, agents operate concurrently into the environment in an asynchronous manner, thus eliminating the typical operation cycles of other Swarm Intelligence systems (e.g., Ant Systems, Particle Swarm Optimization Systems, Cellular Automata, etc.). The general individual behavior of agents is given by Algorithm 4.1.

The crucial step in the algorithm above is the choice of the task which the agent has to execute for the next time step. In its simplest form, it can consist in choosing the one which maximizes the attraction function  $F$ . We will see in the next sections that it can also involve some kind of negotiation between agents and even some kind of local planning.

It is worth to stress the fact that the algorithm allows the agents to respond in real time to changes in the environment like the appearance of new urgent tasks or the temporal impossibility of fulfilling the set of preconditions of a given task.

#### 4.4.4 Inter-agent communication in Bar Systems

Even if Bar Systems don't require from the agents any communicative skills, they are indispensable in order for the system to attain the coordinated and self organized behavior typical of Swarm Intelligence Systems. It is worth

---

**Algorithm 4.1** Individual agents' behavior algorithm in Bar Systems
 

---

```

1: procedure BARSYSTEMAGENT
2:   repeat
3:     Find the most attractive free task M
4:     if the agent is doing M OR trying to fulfill  $pre(M)$  then
5:       Keep doing it
6:     else
7:       Stop doing the current task, if any
8:       if  $pre(M)$  holds then
9:         Start doing M
10:      else
11:        Do some action in order to fulfill  $pre(M)$ 
12:      end if
13:    end if
14:  until no tasks left
15: end procedure

```

---

to differentiate two main classes of inter-agent communicative processes: *direct*, where agents establish direct communication with each other via some channel and following some kind of protocol and *indirect*, where agents communicate with each other through their actions, which cause changes in the environment<sup>3</sup>.

We can identify three main purposes to which communication can serve in order to increase Bar Systems problem solving capabilities:

- *Conflict resolution and negotiation.* The way we defined Bar Systems makes unavoidable the occurrence of conflicting situations in which two or more agents choose the same task to carry out. Lack of communication will lead to a waste of resources because of several agents trying to fulfill the preconditions of the same task, even if only one of them will finally carry it out. In such situations it would be convenient to have some kind of negotiation method which can be as simple as "the first one which saw it goes for it". In section 4.4.4.1, we will discuss a couple of more elaborated negotiation strategies.
- *Perception augmentation.* In the case that agents have limited perception capabilities (we refer to capability to perceive the tasks), communication can allow an agent to transmit to the others information about

---

<sup>3</sup> In the Bar Systems framework, it can be seen as agents generating "communicative tasks" which, when carried out by other agents, increase the information they possess (about the environment, the task set ...)

pending tasks they are not aware of. Let's suppose we want to do some kind of exploratory task in a vast terrain where points of interest must be identified and explored by means of a Bar System. It would be useful that agents had the ability to share information about the points of interest which they have located during their exploratory activity, this way agents would have access to information about the location of points of interest which lie beyond their perceptual capabilities.

- *Learning.* The attraction function  $F$  defined in Section 4.4.1 does not need to be fixed in advance. Agents can learn it through their own activity and their communicative interactions with other agents. For example, an agent can find out that a certain kind of task has a high cost and communicate this fact to the other agents. Not only that, agents can even learn from other agents the way of carrying out new tasks.

#### 4.4.4.1 Negotiation in Bar Systems

As stated before, negotiation plays a key role in Bar Systems performance. Negotiation allow agents to avoid conflicts when two of them share the same most attractive task, helping to decide which agent will take care of it. Several negotiation strategies can be devised when several agents want to do the same task  $t$ , amongst them (from more simple to more complex):

- *FTI.* First Take It. The first agent to choice a given task as its more attractive task carries it out.
- *HP.* Highest Preference. The agent with the greatest affinity (value of the  $F$  function) to task  $t$  carries it out.
- *HPD.* Highest Preference Difference. The agent with the greatest difference between its affinity to  $t$  and its affinity to its second preferred task carries  $t$  out.

In lack of further experimentation, we have the opinion that some kind of combination of the three criteria would be a good option for a wide range of problems.

#### 4.4.5 Local planning in Bar Systems

Although there is nothing like global planning in the way a set of bartenders work, they have tricks that allow them to spare time and effort. For example if two customers are asking for a pint and they are close enough to each other in the bar, the bartender will usually serve them at once. In a similar way, a

taxi driver who is going to pick up a passenger will surely take advantage of the opportunity if he finds in his way a new passenger and he can transport him without deviating too much from his original route. The inclusion of this sort of very simple, problem-dependent, local planning techniques in the choice of the tasks is not difficult and can be done through different methods ranging from local search to the use of expert rules.

## 4.5 The CONTS Problem

A class of problems frequently found in "real life" involves some kind of scheduling in the transport of goods or people from one place to another. The problem which we present as a framework for the study of Bar Systems applicability and efficiency is inspired in the problem which has to be solved by a group of loading robots in a commercial harbor. The task of these robots is to transport the containers from their storage place to the docks where the corresponding ships have to be loaded. Of course, this transport has to be done in such a way that the containers arrive in time to be loaded and with the lowest possible cost. Next we state a formalization (and simplification) of the problem, which we will call CONTS. Afterwards we are going to study its complexity and we will see how we can use a Bar System to solve it efficiently.

### 4.5.1 Definition of the problem

Let  $C = \{c_1, c_2, \dots, c_n\}$  be a set of containers, let  $L = \{l_1, l_2, \dots, l_m\}$  be a set of loading robots and let  $P = \{(x, y) \in \{0..MaxX\} \times \{0..MaxY\}\}$  be a set of positions. Each container  $c_i$  has the following associated properties:

- $p(c_i) \in P$ . The position where the container lies.
- $dest(c_i) \in P$ . The position to which the container has to be carried to.
- $weight(c_i) \in \mathfrak{R}^+$ . The weight of the container.
- $dline(c_i) \in \mathfrak{R}^+$ . The latest instant of time in which the container can arrive to the dock in order to be loaded in time into the ship.

In order not to complicate the problem too much, we will assume that all the containers have the same importance. There are also several properties associated to each loading robot  $l_i$ :

- $p(l_i) \in P$ . The place where the robot is at each instant.
- $maxload(l_i) \in \mathfrak{R}^+$ . The maximum weight the robot is able to carry.
- $maxdist(l_i) \in \mathfrak{R}^+$ . The distance beyond which the robot can't "hear". It allows us to model the perceptual limitations of the robot.

- $speed(l_i) \in \mathbb{R}^+$ . The speed at which the agent can move.

Robots can perform different actions, they can move towards any position, load (if container and robot are in the same position) containers which weigh less or the same as its *maxload* value and download containers.

The problem consists in finding, if it exists, a sequence of actions that allows the robots, departing from their initial positions, to transport every container to its destination point, in such a way that no container arrives after its deadline. In order to simplify the problem, we will assume that the robots always move at the same speed, that uploading and downloading operations are instantaneous and that robots can only carry one container at a time.

#### 4.5.2 Complexity of the CONTS problem

Of course, before trying to solve the problem we have to get an idea of its complexity. Using an heuristic method might not make much sense if there was some exact method of polynomial complexity. On the contrary, if the problem was very complex, using heuristic methods which gave approximate solutions, like Bar Systems, would be justified. The fact is that the problem is not at all trivial. The associated state space is enormous (it is not only necessary to take into account which containers each robot will move and in which order; the solution of some instances of the problem implies moving some containers to a different position from the one of delivery and leave them there to return to take them later) and it is also extremely sensitive to initial conditions, as most of NP-hard problems usually are. In [18] an in-depth study of the problem can be found with a proof of it to be at least as complex as a NP-hard problem. In general terms, the proof reduces the Traveling Salesman Problem (TSP) to CONTS by showing that every instance of the TSP problem is equivalent to an instance of CONTS where there is a single robot and all the containers have the same deadline and have to be delivered in the same position where they lie (please, refer to Appendix D for further information). An exhaustive search method that finds optimal solutions has also been programmed but, as expected, it can only deal with extremely simple instances of the problem.

### 4.6 A Bar System for Solving the CONTS Problem

Once the option of solving the problem in an exact way in the general case has been discarded, we now look at the possibility of using an heuristic method like a Bar System. The idea on which we are going to base it is very simple: to simulate an environment where the containers "shout" to the agents asking for somebody to take them to their destination. The intensity of the shout

of each container depends on the remaining time before its deadline and the distance between its position and the delivery position (it could also depend on the importance of each container, but we must remember that the way we defined the problem, they are all equally important). The robots hear the calls of the containers diminished by the distance, so they go and take the ones they hear better. In order to achieve this behavior in the robots we will use a linear attraction function. Following the notation introduced in section 2, we define, for all container  $c$  and for all robot  $l$ , the attraction function  $F$  in the following way:

$$F(c, l) = \begin{cases} -\infty, & \text{if } c \text{ has been delivered.} \\ -\infty, & \text{if } c \text{ is being delivered for a} \\ & \text{robot other than } l. \\ K_1 \cdot \text{urg}(c) - K_2 \cdot \text{cost}(c, l), & \text{ow.} \end{cases} \quad (4.3)$$

where  $K_1$  and  $K_2$  are adjustable parameters. The urgency function  $\text{urg}(c)$  is defined as inversely proportional to the time which remains to  $c$ 's deadline and takes into account the time required for transporting the container to its destination point:

$$\text{urg}(c) = \text{curtime} + \frac{d(p(c), \text{dest}(c))}{\text{meanspeed}} - \text{dline}(c) \quad (4.4)$$

where  $d$  is the Euclidean distance,  $\text{curtime}$  is the current time and  $\text{meanspeed}$  is an environmental constant which averages agents' speeds. The  $\text{cost}$  function is defined as follows:

$$\text{cost}(c, l) = \begin{cases} \infty, & \text{if } \text{weight}(c) \geq \text{maxload}(l). \\ \infty, & \text{if } d(p(l), p(c)) \geq \text{maxdist}(l). \\ \frac{d(p(l), p(c)) + d(p(c), \text{dest}(c))}{\text{speed}(l)}, & \text{ow.} \end{cases} \quad (4.5)$$

The election of this attraction function  $F$  is quite arbitrary. A non-linear function would probably better reflect the "hearing" metaphor introduced before. In the same way, we could also have defined a more sophisticated urgency function, non-linearly increasing depending on the time to the containers' deadline, for example. Bar Systems are general enough to use any attraction, cost or urgency functions. The question is finding, for each problem, the function which will give the best results. Our choice of the attraction function  $F$  is based in its simplicity, in spite of which, it has allowed us to obtain very good results.

The behavior of the robots will be very simple and it will obey the algorithm described in Section 4.4.3. Each robot will choose a container to go for and will go towards its position, will load it (if not any other robot has arrived first) and will take it to the delivery point. After that, it will repeat the cycle until no containers left to transport.

#### 4.6.1 Inter-agent communication and local planning for the CONTS problem

Aiming to the study of the utility of inter-agent communication, we will investigate two different methods for the choice of the next container to go for. If no communication between agents is allowed, each agent will simply choose the one which maximizes the attraction function. On the other hand, if the possibility of communication between agents is activated, each robot will ask to the others (perhaps not all of them but only those which communication is feasible) which containers they prefer and, in case of conflict (that is, another robot preferring the same container), a small negotiation process will start, the goal of which is to give preference to the agent who will be able to carry faster the container to its delivery position. The agent which finds itself in the situation where other agents have priority over it to transport its favorite container will try with the next best container, in order of preference according to its point of view, until it finds one for which it will have more priority than any other agent. It would be easy to devise more sophisticated negotiation processes taking into account the second-best options of the agents in conflict in such a way that one agent could resign carrying its preferred container, even if it has the higher preference over it, whenever the preference difference between the best and the second-best containers was small enough.

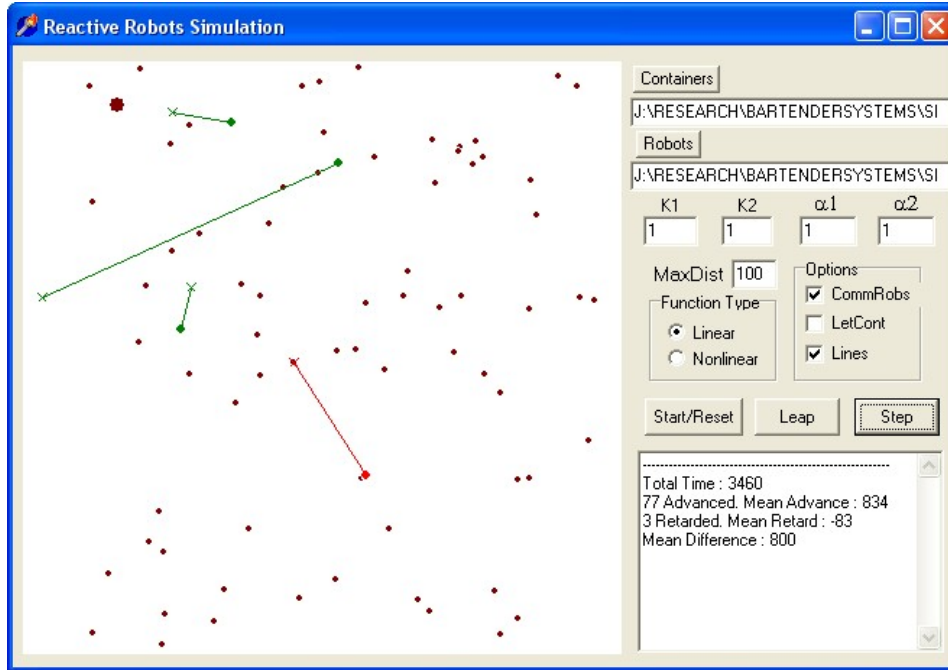
We have also implemented a very straightforward planning-like strategy in our Bar System. Whenever a robot has a container to go for, it looks if there exists another one such that it is possible to transport it without deviating too much from its original way to the first container position. If so, the agent transports it before resuming its original way to the first container position.

### 4.7 Results

In order to analyze the efficiency of our method and experiment with different settings and parameter values, we have programmed a graphical simulator for the problem (see Figure 4.1). We have chosen an instance of the problem with eighty containers randomly scattered on a  $300 \times 300$  rectangular area with random delivery points and deadlines and four carrier robots, all of them with the same value for the parameter *maxdist* and different speeds. We have done two main sets of simulations experimenting with different values of the parameters *K1* and *K2*. In the first set (figure 4.2) we don't allow agents to communicate or perform any local planning, whereas in the second set (figure 4.3) communication and local planning are permitted.

We can see in figures 4.2 and 4.3 the results of the two sets of simulations.

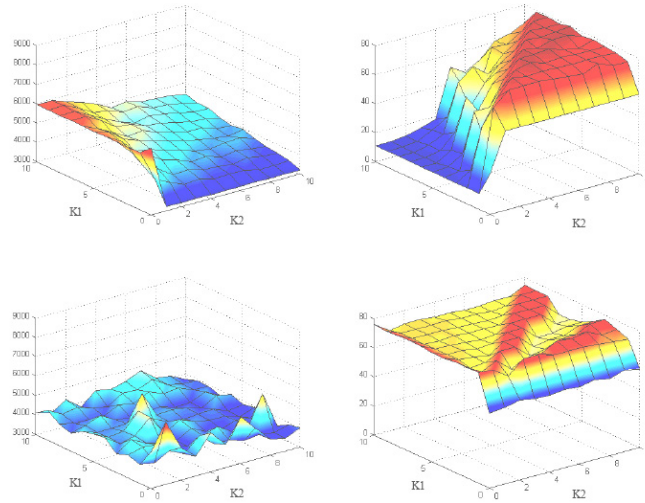




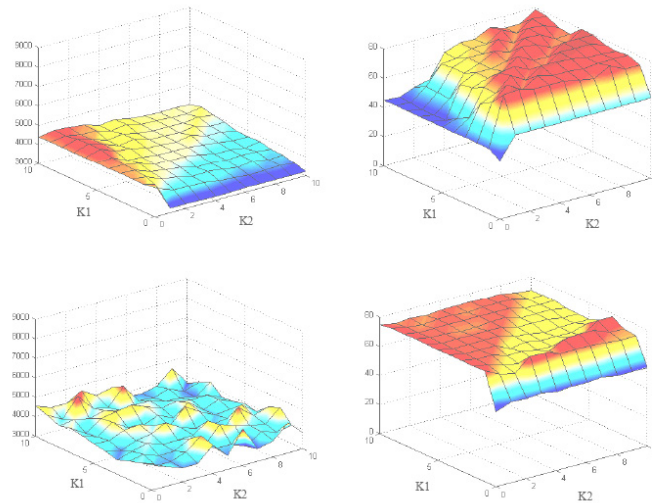
**Figure 4.1:** Screen capture of the simulation program in action. Lines indicate the destination point of each robot. Red lines correspond to carried robots heading towards the container's delivery point. Green lines correspond to free robots making their way towards their preferred containers. Big dots indicate timed-out containers.

Each row represents a series of 121 simulations (for values of the  $K1$  and  $K2$  parameters ranging from 0 to 10 in increases of 1). The charts in the left columns show the time used to deliver all the containers and the charts in the right columns show the number of containers delivered before their deadlines. The two rows correspond to different values (300 and 100) of the *maxdist* parameter.

We can draw several conclusions. On one hand, it is clear that, for some values of the parameters  $K1$  and  $K2$ , the system finds much better solutions than those which can be obtained by using nearest neighbor-like methods. We can observe the performance of those methods in the top row of figure 4.2, when  $K1 = 0$  the preference function  $F$  depends only on the *cost* function and the systems behaves in the “nearest container” way. The results are a low total delivery time and a considerable number of containers delivered after its deadline. The case  $K2 = 0$  is even worse. The system follows the “most urgent container” behavior, resulting in very long displacements which cause a big total delivery time and ,consequently, a big number of containers delivered with retard. It is worth to remark that the improvement over



**Figure 4.2:** *Left: Total time needed by the system to deliver all the containers for different values of the parameters  $K1$  and  $K2$  and for different values of the parameter  $maxdist$  (top row  $maxdist = 300$ , bottom row  $maxdist = 100$ ). Right: Number of containers delivered before their deadlines. Communication and local planning are deactivated.*



**Figure 4.3:** *Left: Total time needed by the system to deliver all the containers for different values of the parameters  $K1$  and  $K2$  and for different values of the parameter  $maxdist$  (top row  $maxdist = 300$ , bottom row  $maxdist = 100$ ). Right: Number of containers delivered before their deadlines. Communication and local planning are permitted.*

those greedy methods achieved by our Bar System for some values of the parameters  $K1$  and  $K2$  is not attained in exchange of a greater complexity; in fact, the complexity of the system, understood as the amount of work which each agent has to do in order to decide the next container to go for, increases lineally with the number of containers.

We can also observe how the quality of the solutions found depends on the perceptual capabilities of the agents. When this capability is very limited (not shown in the figures), robots' behavior is too local, resembling somewhat like a mixture of "nearest container" and random walk. On the other side, very good solutions are found for certain values of the parameters  $K1$  and  $K2$  when the agents are able to perceive the environment almost entirely ( $maxdist = 300$ ). This augmented perceptual ability implies, nevertheless, the possibility of appearance of several phenomena which can affect system's efficiency, like, for instance, that it will be necessary to evaluate more alternatives, that the probability of conflicts will increase and that, depending on the values of the parameters, the system can arrive to very bad solutions if the agents must perform long displacements. Thus, a bit paradoxically, more perception power can yield poorer results. The most interesting case, from our point of view, is when the agents have a perceptual capability between the two extreme points. We have tested the case  $maxdist = 100$  and we can see in the bottom row of figure 4.2 how the system finds good solutions for most values of the parameters  $K1$  and  $K2$ . There are particularly, two big zones in the parameters space where the solutions found are as good as the ones obtained by the agents of the first row of the figure, which have a perceptual power nine times greater.

In figure 4.3, we can see how the inter-agent communication or negotiation and local planning can improve greatly, depending on the values of the parameters, the quality of the solutions found. Clearly, the importance of communication between agents increases with the possibility of conflict, which is proportional to the agents' perception power and decreases with the relative magnitude of the parameter  $K2$  regarding  $K1$ . The more  $K2$  grows regarding  $K1$ , the more importance is given to the distance to the container in the calculation of the preference function, the robots tend to prefer the nearest containers and the number of conflicts decrease, as well as the utility of communication.

It is interesting to note that for some values of the parameters  $K1$  and  $K2$ , communication and local planning capabilities doesn't improve system's results. This is probably due to the fact that the results for those parameter values in the Bar System without communicative or planning capabilities are near-optimal (all the containers delivered in time). Nevertheless, it is clear that more work in this direction is needed in order to clarify communication and local planning effects.

## 4.8 Conclusions and Future Work

We have presented Bar Systems, a new class of reactive multi-agent systems for distributed problem solving. They are loosely inspired in the way a group of waiters work behind a bar. We have formalized the definition and we have given the general algorithm which rules the behavior of the individual agents. Several of the characteristics of Bar Systems are:

- **Simplicity.** Agents in Bar Systems are simple. They share a similar structure and operate in a decentralized manner in a similar way optimizing a local function. Interactions between agents are simple, too.
- **Efficiency.** Bar Systems have lineal complexity with respect to the number of tasks.
- **Robustness.** Faults in individual agents do not decrease dramatically the efficiency of the system. Moreover, Bar Systems' problem solving capabilities increase steadily with the addition of new agents.
- **Responsiveness.** Bar Systems respond easily to the occurrence of unforeseen events as the appearance of new high priority tasks.

All those characteristics, jointly with the capability to seamlessly integrate different more or less sophisticated negotiation and local planning techniques, make Bar Systems very suitable to solve problems in asynchronous, dynamical, partial information and time-critical environments.

To check the efficiency and applicability of Bar Systems, we have defined a NP-Hard problem called CONTS, based on the work which a set of robots has to perform to transport a set of containers in time to their destination. The Bar System used to solve it has proved to give much better results than other greedy algorithms of the nearest neighbor type and has established, in our opinion, the usefulness of Bar Systems as a general framework for solving this type of real time problems. We have also seen that communication amongst agents and local planning allows improving the results greatly without increasing the complexity of the system significantly.

Our work in Bar Systems is just starting and we are aware that there are many aspects that require more study and testing. Some of the directions in which it would doubtless be worth working and which essentially refer to the nature of the attraction function  $F$  are:

- Study of Bar Systems' performance in highly dynamical, time-critical environments. We are currently considering its use in the ROBOCUP and RESCUE environments.
- Study of the applicability of Bar Systems to other kinds of problems. At a first glance it could seem Bar Systems to be a bit too restrictive with

respect to the kind of problems which they can tackle, We must remark, nevertheless, that its application is not limited to problems involving the transportation of goods or people (and we don't mean it to be a narrow application field, it is wide enough to contain problems ranging from service assignation in cab companies to multiagent autonomous terrain exploration), they can also be useful in other problems which do not necessarily involve physical movement of the agents (actually, there is a considerable amount of work in progress [17] concerning the applicability of Bar Systems to digital document preservation, for example). They are also applicable to resource allocation problems in the style of flexible Open-Shop problems where the order in which a set of machines, in a factory, for instance, has to perform a set of operations has to be decided. Moreover, with an appropriate definition of the attraction function  $F$ , a Bar System can be used for solving flexible Job-Shop problems, where there is a set of independent jobs, each formed by a series of tasks which have to be done in a specific order. In this kind of problems, for each job there is at all times, at the most, just one feasible task, and it would be sufficient to define the attraction functions in such a way that all job's not done tasks "transmit" their urgency to the feasible one. The same idea could be used in a more general setting, where there would simply be any type of non-cyclical precedence relations over the set of tasks. It can also be worth to study the applicability of Bar Systems in competitive environments.

## Chapter 5

# Using FCFs to Filter Inter-Agent Communication in Bar Systems

### 5.1 Introduction

We have seen in Chapter 4 how Bar Systems can show remarkable performance in the resolution of difficult problems like CONT and how this good performance relies heavily upon the communication abilities of the individual agents. The importance of the inter-agent communication makes possible, therefore, that the failure of one or more agents to provide true and accurate information to the other agents in the system could result in a substantive deterioration of the problem solving capability of the Bar System.

Main causes for the communication of false or unaccurate information were exposed in Section 1.2. They can be divided in two classes: intentional and unintentional. Intentional false communication may occur typically in competitive scenarios where an agent tries to gain a competitive advantage by fooling their more collaborative mates. Unintentional communication of false information, on the other hand, can be due to several reasons. For example, the agent that emits the information can speak a different language that the agents that receive it (perhaps because they are using different measurement systems). It is also possible for an agent to be not aware of some disfunctionality in its behaviour which makes it to communicate inaccurate information (a defective sensor, for example). Whichever the case, it is clear that a mechanism able to deal with those problems is needed.

We saw in the ammeter case study in Chapter 2 how FCFs can deal effectively with the problem of unintentional false information, the kind of

false information expected to appear in collaborative environments. In this chapter we will show how FCFs can also deal with the problem of the communication of intentionally false information in a competitive environment. We will define a scenario where several taxi driver agents will compete for customers and we will show how FCFs can be used to decide whether to compete or not against the other agents for a customer. A quite simple evolutionary method will show how individual agents can improve their performance by refining its internal models (the FCFs) of the competitor agents.

The structure of the paper is as follows: Section 5.2 formalizes the problem and briefly describe the simulator and the evolutionary algorithm used to obtain the results, which are commented in Section 5.3. Section 5.4, finally, presents the conclusions and enumerates several possible directions for the future work.

## 5.2 Case Study. Competitive taxi driver agents

This section defines and formalizes the competitive taxi driver agents problem and describes the implementation of a simulation that uses it as a workbench for FCF testing.

### 5.2.1 The problem

The problem can be summarized as follows: the job of a set of taxi driver agents involves the transport of passengers, in exchange of money, between different points in a territory. Passengers can appear at random positions and times and have an associated profit, the amount of money paid to the agent that transports him to its destination point. The agents must compete between them in order to achieve the maximum total profit. Let's formalize it a bit more:

Let  $P = \{(x, y) | x \in [1 \dots MaxX], y \in [1 \dots MaxY]\}$  be a set of positions and let  $A = \{a_1 \dots a_m\}$  be a set of taxi driver agents. We define, for each taxi driver  $a_i$  the following quantities:

- *Position*  $p_i = (p_{i_x}, p_{i_y}) \in P$ . The current position of taxi driver  $a_i$ .
- *Speed*  $s_i \in \mathfrak{R}^+$ . The displacement speed of taxi driver  $a_i$ . We assume it constant, in order to simplify the problem.

On the other hand, let  $C = \{c_1 \dots c_n\}$  be a set of customers. Each customer  $c_i$  has the associated quantities:

- *Position*  $p_i = (p_{i_x}, p_{i_y}) \in P$ . The current (origin) position of passenger  $c_i$ .

- *Destination*  $d_i = (d_{i_x}, d_{i_y}) \in P$ . The position passenger  $c_i$  wants to go.
- *Profit*  $Q_i \in \mathbb{R}$ . The amount passenger  $c_i$  is willing to pay for the trip.

Taxi driver agents can perform two main actions: The first is to go to any place in the territory (usually to load a passenger, although random moves are allowed when no passenger is in sight). The second action is to transport passengers to their destination points. Passengers pay to the taxi driver in the moment of the arrival. For the sake of simplicity, we will suppose that:

- Passenger loading and unloading are instantaneous processes.
- All taxi driver agents can carry any passenger, but only one at once.
- Taxi driver agents can calculate and express the interest they have in each individual passenger. This number somehow reflects the extent to what they are eager to compete for it. Agents know the interest the other agents have in each passenger and can use this information to decide whether to compete for them or not. However, agents can lie about the interest they have in order to discourage competitors.

Individual agents face then with the problem of finding the strategy that maximizes their profit in this competitive environment.

### 5.2.2 The simulation

---

**Algorithm 5.1** Main procedure of the simulator

---

```

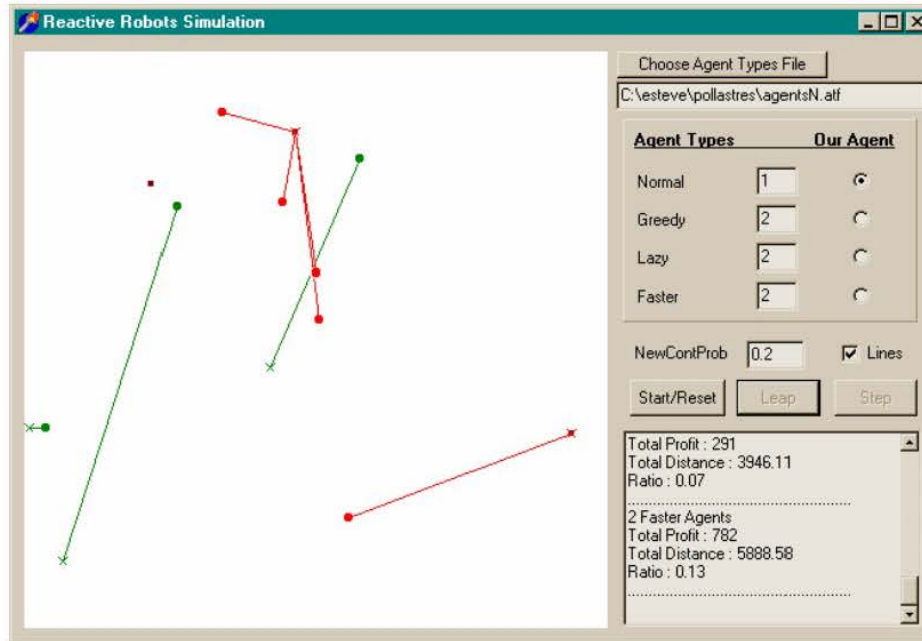
1: function SIM.RUNSIM : returns results;
2:   Sim.Initialize;
3:   repeat
4:     Sim.lookForNewPassengers(NewPProb);
5:     for all agent a in Sim.agents do
6:       a.chooseDirection;
7:       a.updatePosition;
8:       a.doThings;
9:     end for
10:    Sim.paint;
11:   until sim.finished;
12:   return Sim.statistics;
13: end function

```

---

A simulator has been developed several series of experiments have been done. Algorithm 5.1 shows a pseudo-code representation of the main procedure. After a first initialization part used to load the input files and parameters, the program enters in the main loop. Each iteration has three



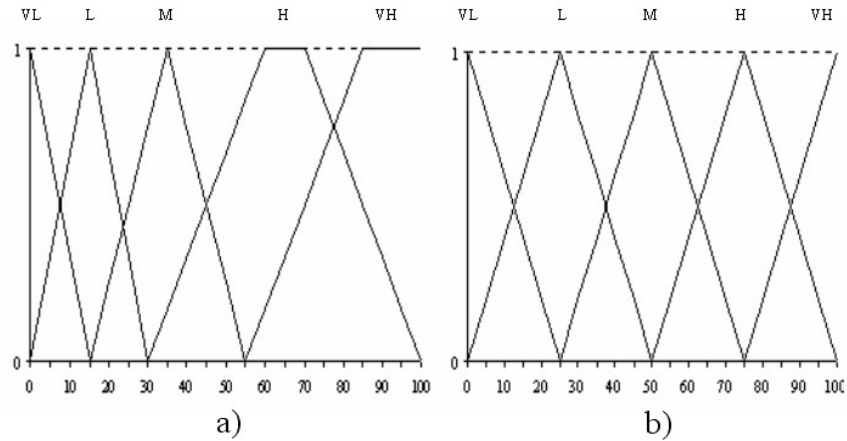


**Figure 5.1:** *The graphical interface to the simulation. Agents (big dots) go for passengers (small dots) and then transport them. In the figure, we can see four agents competing for the same passenger.*

parts. First part randomly generates, according to a fixed probability distribution, new passengers with their associated profit. The second part calls, for each agent, the three subroutines that define its behavior: *chooseDirection* chooses a direction to go based on the available passengers and the information provided by other agents about their preferences, *updatePosition* simply calculates the position of each agent at each simulation cycle. Finally, the procedure *doThings* performs, if necessary, the load and unload actions. The third part of the main procedure of the simulator actualizes the graphic output (Figure 5.1). Finally, when the simulation is finished, (usually after a prefixed number of iterations) the program calls a procedure that computes some statistics and results about the agents' behavior and performance.

### 5.2.3 The fit function. Computing the preferred customer

The precedent section showed how, at each iteration of the main loop of the simulator, agents must decide the best direction to go. That decision is quite simple to take: if the agent is currently carrying a passenger, he keeps his current direction towards the passenger's destination point. If the agent carries nothing, a fit function is evaluated for each available passenger and the agent choose as current direction the position of the passenger that



**Figure 5.2:** Fuzzy sets corresponding to the linguistic labels associated to (a) the input variable distance and (b) the input variable profit and the output variable interest.

maximizes it. After that, the agent will decide whether to compete or not against the other agents that prefer the same passenger.

The fit function takes two parameters: the distance from the agent to the current position of the passenger and the profit associated to him. The returned value must represent the preference degree or interest that the agent has for this individual passenger. Each agent can have a different fit function, and therefore two different agents can prefer different passengers even if the distance to the passengers is the same for both.

Fit function computation has been chosen to be implemented by means of fuzzy systems. There will be three fuzzy variables: *distance*, *profit* (as input variables) and *interest* (as output variable), each of them taking values over the fuzzy power set of  $\{0 \dots 100\}$ . Also, for each variable, a set of linguistic labels  $D = \{VeryLow, Low, Medium, High, VeryHigh\}$  is defined (their meaning can be seen in Figure 5.2). The fuzzy rules forming the fuzzy systems have then a form similar to this one:

**If distance is High and profit is Low then interest is Low**

The application of the reasoning procedure to the inputs and the rule base gives as a result a fuzzy set as the value of the output variable, *interest*. A defuzzification procedure will then be applied to this resulting fuzzy set in order to obtain a crisp value between 0 and 100. This value is used to establish preferences upon the set of available passengers.

This formalism is quite declarative and easy to understand, and that is powerful enough to allow the definition of a broad range of different behaviors

in the agents.

#### 5.2.4 Filtering the fit values provided by the agents

Once agent  $A$  has chosen the passenger he prefers to go for (let's name it  $C$ ), it must look if there are other agents interested in the same passenger. The other agents interested in  $C$  are currently going towards it and it's not possible for  $A$  to know things like the distance from the other agents to the passenger or the speed of the other agents. However,  $A$  has to take the decision of whether to compete or not for  $C$ . That decision must be based in the only piece of information provided by the other agents, their interest for  $C$  (a number between 0 and 100). It's clear that, if some other agent shows an interest much higher than that of  $A$ , then the better strategy for  $A$  is to renounce to  $C$  and to choose another less popular passenger. However, it's very possible that a simple comparison between the interest values provided by the other agents and its own interest would not be the best choice for  $A$ . Other agents can, consciously or not, provide false information (from the point of view of  $A$ ). For example, greedy agents can express their interest depending almost only on the profit associated to the passengers, that is, they can manifest a high interest for profitable passengers even if they are very far from them. It is also possible for the two agents not to speak the same language, so giving different meanings to the same interest values. In any case, it seems a sensible strategy for  $A$  to somehow filter the interest values provided by the other agents agents, diminishing or augmenting them depending on whether they refer to more or less profitable passengers, before comparing them with his own interest in order to decide for competition.

In our system, that task is accomplished by means of FCFs that filter the interest manifested by the other agents before comparing with  $A$ 's own. The posterior decision mechanism is very simple: if  $A$ 's interest is greater than the filtered interest values of all the other agents interested in  $C$  then  $A$  will compete for  $C$ , else  $A$  will look for another passenger. Each taxi driver agent will have, thus, a FCF for each of their competitors (In fact, there is a set of classes of taxi driver agents, so that each agent only has to maintain a FCF for each class).

Let  $A$  and  $B$  be agents and let  $C$  be a passenger. Let  $i(B, C) \in \{0 \dots 100\}$  be the interest in passenger  $C$  manifested by  $B$ .  $A$ 's very simple FCF for  $B$  will have only one input fuzzy variable, called *interest* (the main variable), and an output fuzzy variable, called *finterest* (the filtered variable), taking both of them values over the fuzzy power set of the set  $\{0 \dots 100\}$ . No side variables are included in the filters (although the use of the profit as the fuzzy side variable could probably lead to better results) for the sake of performance of the evolutionary algorithm used in the experiments.

Values for the fuzzy input variable are obtained by means of a fuzzy-

fication process consisting in considering as a singleton fuzzy set the crisp value  $i(B, C)$ . There are also defined, for each variable, a set of linguistic labels  $D = \{VeryLow, Low, Medium, High, VeryHigh\}$  (the forms of the corresponding fuzzy sets are the same that the ones found on the right part of Figure 5.2. So, the five rules in each fuzzy filter have a form similar to this one:

**If interest is VeryHigh then finterest is High**

As it was in the case of the fuzzy systems used to compute the interest that each agent has for a passenger, the application of the fuzzy reasoning procedure results in a fuzzy set as the value of the output variable, *finterest*. We can again apply a defuzzification procedure to obtain a crisp value between 0 and 100. That's the value that *A* will compare with his own interest for the passenger *C* to decide whether to compete or not for him with *B*.

### 5.2.5 A simple evolutionary algorithm

It is very easy to use the simulator to experiment with different agent types. For example, more or less greedy or lazy agents (depending on if its main motivation is profit or distance) can be defined, as well as more or less quick ones. It is important to remark that agents can differ in two main ways. On one hand, they may differ in the rules used to assign interest to the passengers; some agents can prefer the nearer ones, some the more profitable. On the other hand, it can exist a deeper difference in the way in which the agent view the world. For example, the meaning of the label *Low* can be different for different agents if so are the fuzzy sets defining it. That means that agents with the same set of rules can behave quite differently.

In order to carry out our experiments, a simple evolutionary algorithm has been used (see Algorithm 5.2) in which a single agent among a multiagent community is allowed to evolve, adapting its FCFs to improve its performance (the total profit per simulation run) It is important to note that the fuzzy systems that compute interests remain constant along the simulation cycles, only FCFs evolve. The operation of the algorithm is as follows: after a first step of initialization in which the agents (several of each type usually) are created and positioned, a first run of the simulation is carried out and a first result is obtained, (simply the total profit obtained by the only evolving agent). After that, the program enters the main loop, which is executed a prefixed number of iterations. Each iteration has three main parts. First, the fuzzy filters of the evolving agent are mutated randomly. Second, a new simulation is run and a new result is obtained. Finally, this new result is compared with the old one. If the new result better, then the value of the result variable is actualized, else the mutations are undone before proceeding

with the next iteration. The hope is the evolving agent to gradually increase its performance by adapting itself to the other agents.

---

**Algorithm 5.2** Simple evolutionary algorithm

---

```

1: function SIM.SIMPLEEVOL : returns results;
2:   Sim.Initialize;
3:   Result  $\leftarrow$  Sim.runSim;
4:   Iteration  $\leftarrow$  1;
5:   repeat
6:     inc(Iterations);
7:     Sim.makeMutation;
8:     NewResult  $\leftarrow$  Sim.runSim;
9:     if NewResult is better than Result then
10:      Result  $\leftarrow$  NewResult;
11:    else Sim.undoMutation;
12:    end if
13:  until Iterations=MAXITERATIONS;
14:  return Result;
15: end function

```

---

### 5.3 Results

Four main types of agents were used in the experiments:

1. *Lazy*. The preference of this type of agents for the passengers is almost exclusively calculated from the distance to them. They tend to prefer nearer passengers even if there are much more profitable passengers a little bit far away.
2. *Greedy*. This type of agents assign preference to the passengers depending mainly on its associated profit. They prefer more profitable passengers even if there is a large distance to them.
3. *Smart*. This type of agents compute their interest for the passengers attaching similar significance to profit and distance.
4. *Fast*. The set of rules for those is the same than that of smart ones. The difference between the two types is that fast agents have a speed of 15 while smart agents (as lazy and greedy agents) have a speed of 10.

Figure 5.3 shows the matrices representing the fuzzy systems used for interest calculation associated with each agent type. On the other hand, we

	VL	L	M	H	VH
VL	VH	H	M	L	VL
L	VH	H	M	L	VL
M	VH	H	M	L	VL
H	VH	H	M	L	VL
VH	VH	H	M	L	VL

a)

	VL	L	M	H	VH
VL	VL	VL	VL	VL	VL
L	L	L	L	L	L
M	M	M	M	M	M
H	H	H	H	H	H
VH	VH	VH	VH	VH	VH

b)

	VL	L	M	H	VH
VL	M	L	VL	VL	VL
L	H	M	L	L	VL
M	VH	H	M	L	VL
H	VH	H	H	M	L
VH	VH	VH	VH	H	M

c)

	VL	L	M	H	VH
VL	M	L	VL	VL	VL
L	H	M	L	L	VL
M	VH	H	M	L	VL
H	VH	H	H	M	L
VH	VH	VH	VH	H	M

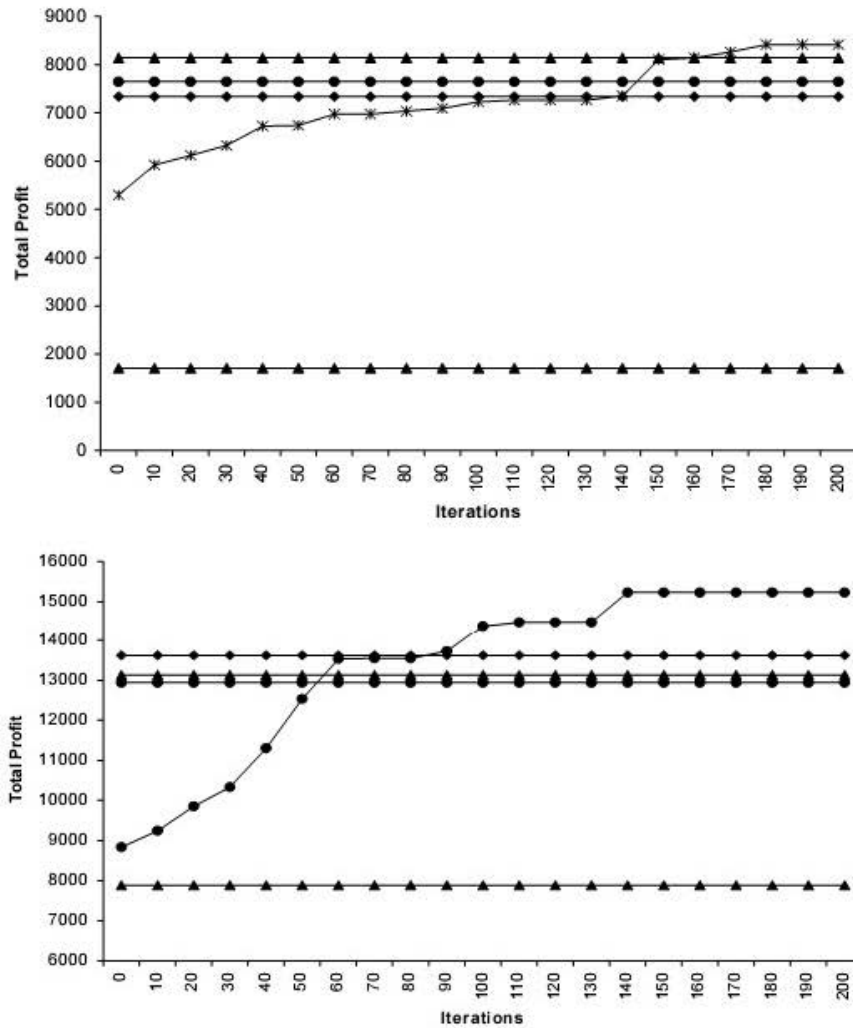
d)

**Figure 5.3:** The matrices representing the fuzzy systems used by each agent type to assign interest values to the passengers. Rows represent the fuzzy values of the variable profit and columns those of the variable distance. (a) Lazy agent. (b) Greedy agent. (c) Smart agent. (d) Fast agent. Note that smart and fast agents share the same matrix.

have to define the sets of fuzzy sets that give meaning to the linguistic labels for each of the fuzzy variables: *distance*, *profit*, *interest* and *finterest*. For the sake of simplicity, we have decided to have two common sets of fuzzy sets for all four agent types. That corresponds to a MAS system in which all the individual agents share a similar vision of the world. That is, all them mean the same thing when they say, for example, that the distance to a passenger is very high or that the profit associated with a passenger is medium. The definition of the labels can be seen in Figure 5.2. In the experiments, a total of five simulations were carried out, each of them consisting of 200 iterations of 12000 cycles. In all the simulations, a single evolving smart-type agent competes against six other agents. In the first four simulations, competitors are all of the same type (smart, greedy, lazy and fast, respectively). In the fifth simulation, competitors are one of each type. The results of the five series are quite similar. Figure 5.4 shows a summary of the results of two of the simulations. In both parts of the figure, a graph of the performance of the agent without the use of FCFs, as well as several graphs of its performance using specific, constant FCFs are shown for reference purposes. From the comparison between the performances of agents using FCFs and that of those that do not use them, it is evident that FCFs have the capability of significantly increase the performance of the agents. On the other hand, it is also evident that FCF learning is perfectly possible (even with an algorithm as simple as the one used). This feature of FCFs, along with their simplicity and generality, make them an excellent framework for the modeling of some social aspects, like trust and confidence, of MAS.

## 5.4 Conclusions and Future Work

In this chapter we have discussed the role that information filtering plays in both collaborative and competitive multiagent systems. We have reached the conclusion that it is necessary for an agent in a MAS, in order to increase its performance, to have mechanisms allowing it to model the other agents from the point of view of the confidence they deserve. We have presented a method based on fuzzy set theory that allows the filtering of the information provided by other agents through the use of FCFs. This method is simple, general and suitable to the application of a wide variety of learning techniques. We have presented, too, a simulation in which a community of taxi driver agents compete against each other for transporting passengers and we have, finally, seen how social knowledge, understood as knowledge about the rules that determine the behavior of other agents in the community, can be learnt by individual agents, in the form of fuzzy rules in FCFs, through evolution. There are several directions in which this research can be extended. In particular, the following subjects can be object of further study:



**Figure 5.4:** Above, a single evolving smart agent with FCFs competes against a population of simple smart agents. Below, the same agent competes against greedy agents. In both figures, constant graphs correspond to the profit attained by the agent by using, from top to down, a hand-crafted fuzzy filter, a fuzzy filter that enforces competition, no fuzzy filter at all, and a fuzzy filter that avoids competition. The non-constant graphs reflect the increment of the performance of the agents using evolving fuzzy filters.



- The performance of FCFs as modelling tools when there is a whole population using them.
- How evolution of the sets defining the linguistic labels of a FCF affect its performance.
- The application of different learning techniques to the social learning process. Reinforcement learning techniques and genetic algorithms seem to be valid options.
- The identification of new fields suitable to the application of FCFs.

## Chapter 6

# Conclusions

This thesis introduces the novel (concerning the multiagent systems field) concept of reliability. It argues that reliability not only extends the concept of trust, but also beats it in terms of usefulness. Reliability is defined as the quality of an agent that tends to give the same or a similar response when the same or a similar question is asked under the same or similar circumstances, regardless of the erroneous the answer can be. This regularity in the error committed by the agent in his response can allow us to extract knowledge from it.

The device presented in order to make possible the computation of the reliability of a given agent under a given set of circumstances, as well as the correction of the error committed by the agent, is called a Fuzzy Contextual Filter (FCF). FCFs are composed by two parts: a corrective module, that given an input value provided by an agent and a set of values for other variables in the environment, filters the input value (that is, tries to remove the error from the input value) to obtain a new, filtered value, and a second module, the reliability computation module, which computes the confidence we can deposit in the filtered value provided by the corrective module. The corrective module of a FCF consists basically in a set of rules forming a fuzzy system. Those rules are obtained from experience or common sense and from past interactions between the agent possessing the FCF and the agent whose output is to be filtered. Reliability is computed as a combination of three fuzzy system's quality criteria applied to the fuzzy rule bases in the corrective module: completeness, redundancy and consistency. This value can be computed exclusively from the structural characteristics of the system. That is, from the form of its rules and the fuzzy sets used in their definition. The computation of this quality value, then, is easily automatable. A experiment consisting in a simulation of the behavior of an ammeter under different conditions of input intensity and temperature has been carried out. The results

of the experiment confirm the applicability of FCFs to this type of problems.

Chapter 3 has presented several aspects of the ART Testbed and introduced the techniques and algorithms behind the appraiser agent SPARTAN, specially the application of FCFs to the estimation of other agents's appraisals variances.

Chapter 4 has introduced Bar Systems, a new class of reactive multi-agent systems for distributed problem solving. They are loosely inspired in the way a group of waiters work behind a bar. Several of the characteristics of Bar Systems are:

- **Simplicity.** Agents in Bar Systems are simple. They share a similar structure and operate in a decentralized manner in a similar way optimizing a local function. Interactions between agents are simple, too.
- **Efficiency.** Bar Systems have lineal complexity with respect to the number of tasks.
- **Robustness.** Faults in individual agents do not decrease dramatically the efficiency of the system. Moreover, Bar Systems' problem solving capabilities increase steadily with the addition of new agents.
- **Responsiveness.** Bar Systems respond easily to the occurrence of unforeseen events as the appearance of new high priority tasks.

All those characteristics, jointly with the capability to seamlessly integrate different more or less sophisticated negotiation and local planning techniques, make Bar Systems very suitable to solve problems in asynchronous, dynamical, partial information and time-critical environments.

To check the efficiency and applicability of Bar Systems, a NP-Hard problem called CONTS has been defined, based on the work which a set of robots has to perform to transport a set of containers in time to their destination. The Bar System used to solve it has proved to give much better results than other greedy algorithms of the nearest neighbor type and has established, in our opinion, the usefulness of Bar Systems as a general framework for solving this type of real time problems. We have also seen that communication amongst agents and local planning allows improving the results greatly without increasing the complexity of the system significantly.

Finally, Chapter 5 has discussed the application of FCFs to multiagent competitive environments by means of a Bar Systems simulation in which a community of taxi driver agents compete against each other for transporting passengers. The simulation has shown how social knowledge, understood as knowledge about the rules that determine the behavior of other agents in the community, can be learnt by individual agents, in the form of fuzzy rules in FCFs, through evolution.

## Part III

# Appendices



# Appendix A

## Fuzzy Sets and Fuzzy Systems

This appendix gives several definitions on Fuzzy Sets and Fuzzy Systems theory. It by no means pretends to be exact or complete, but only to serve as a quick and basic reference to the reader. A detailed explanation of fuzzy sets, fuzzy logic and fuzzy inference systems theories is beyond the scope of this appendix, we refer the interested reader to [56, 61] and specially to [38], where several excellent introductory chapters can be found.

### A.1 Definitions on Fuzzy Sets

**Definition A.1** *Let  $X = \{x_1..x_n\}$  be an universal set. Every function  $m_F : X \rightarrow [0..1]$  defines a fuzzy set  $F$  over the universal set  $X$ . Function  $m_F$  is called the **membership function** of the fuzzy set  $F$ . For all element  $x \in X$ ,  $m_F(x)$  expresses the degree to wich  $x$  belongs to  $F$ .*

**Definition A.2** *Given two fuzzy sets  $F$  and  $G$  over two universal sets  $U_F$  and  $U_G$  defined by the membership functions  $m_F$  and  $m_G$ , we define the membership functions corresponding to the intersection, union and complementation of  $F$  and  $G$  as follows <sup>1</sup>:*

$$\begin{aligned}m_{F \cap G}(x) &= \min(m_F(x), m_G(x)) \\m_{F \cup G}(x) &= \max(m_F(x), m_G(x)) \\m_{F^c}(x) &= 1 - m_F(x)\end{aligned}\tag{A.1}$$

---

<sup>1</sup>Those are the so-called classical fuzzy set operators, there are other ways to define reasonable and consistent operations on fuzzy sets. See [38] for an in-depth treatment

**Definition A.3** Let  $X$  be some universal set, we will call the **complete fuzzy subset** of  $X$  to the fuzzy set  $C_X$  given by the function  $m_{C_X}(x) = 1$  for all  $x \in X$ . Similarly, we define the **empty fuzzy subset** of  $X$  as the fuzzy set  $\emptyset_X$  given by the function  $m_{\emptyset_X}(x) = 0$  for all  $x \in X$ .

**Definition A.4** Let  $X$  be some universal set. Let  $F$  and  $G$  be two fuzzy subsets of  $X$ . We will say that  $F$  is contained in  $G$  or that  $F$  is a subset of  $G$  ( $F \subseteq G$ ) if and only if  $m_F(x) \leq m_G(x)$  for all  $x \in X$ .

**Definition A.5** Let  $X$  be some universal set, we define  $2^X$ , the fuzzy power set of  $X$ , as the set of all the fuzzy subsets of  $X$ .

**Definition A.6** Let  $X = \{x_1..x_n\}$  be an universal set. Let  $F$  be a fuzzy subset of  $X$ . We define the cardinality of  $F$  as :

$$M(F) = \sum_{i=1}^n m_F(x_i) \quad (\text{A.2})$$

so,  $M(C_X) = n$ ,  $M(\emptyset_X) = 0$  and, in general  $0 \leq M(F) \leq n$  for all fuzzy subset  $F$  of  $X$ .

## A.2 Fuzzy Systems

The fuzzy inference system (fuzzy system for short) is a popular computing framework based on the concepts of fuzzy set theory, fuzzy if-then rules and fuzzy reasoning. It has found successful applications in a wide rank of fields, such as automatic control, data classification, decision analysis, expert systems, time series prediction, robotics, and pattern recognition. A collection of papers covering a broad range of applications of fuzzy inference systems can be found in [4]. The basic structure of a fuzzy system consists of three conceptual components:<sup>2</sup>

- A **rule base**, which contains a set of fuzzy rules of the form:

**if  $A_1$  is  $S_1$  and  $A_2$  is  $S_2$  and ... and  $A_n$  is  $S_n$  then  $B$  is  $T$**

---

<sup>2</sup>There are other several different types of fuzzy systems, attending mainly to the form of the rules and the reasoning mechanism. The definition stated above stands for the so-called Mamdani fuzzy inference systems. However, it is possible to define fuzzy filters based on other types of fuzzy inference systems.

where  $S_1, S_2, \dots, S_n$  and  $T$  are linguistic values, defined by fuzzy sets on universes of discourse  $X_1, X_2, \dots, X_n$  and  $X_T$  respectively, and  $A_1, A_2, \dots, A_n$  and  $B$  are fuzzy variables taking values over the fuzzy power sets of the same universes. The  $A_i$ 's are the input fuzzy variables and  $B$  is the output fuzzy variable. An example of such a rule could be:

**if flow is High and level is High then valve is VeryOpen**

- A **database** which defines the meanings, in the form of membership functions, of the fuzzy sets used in the fuzzy rules.
- A **reasoning mechanism** or **fuzzy inference engine** which performs the inference procedure upon the rules and given facts (values of the input fuzzy variables) to derive a reasonable output or conclusion. That conclusion will take the form of a fuzzy set over  $X_T$

It's worth to note that, although the inputs and outputs of a fuzzy system are, by the above definition, fuzzy sets, it is possible for a fuzzy system to operate with crisp inputs values by applying a fuzzyfication procedure (consisting, for example, in considering crisp inputs as singleton fuzzy sets). In a similar way, several different defuzzyfication mechanisms can be applied to the fuzzy set obtained as conclusion by the fuzzy system in order to obtain a crisp output value.



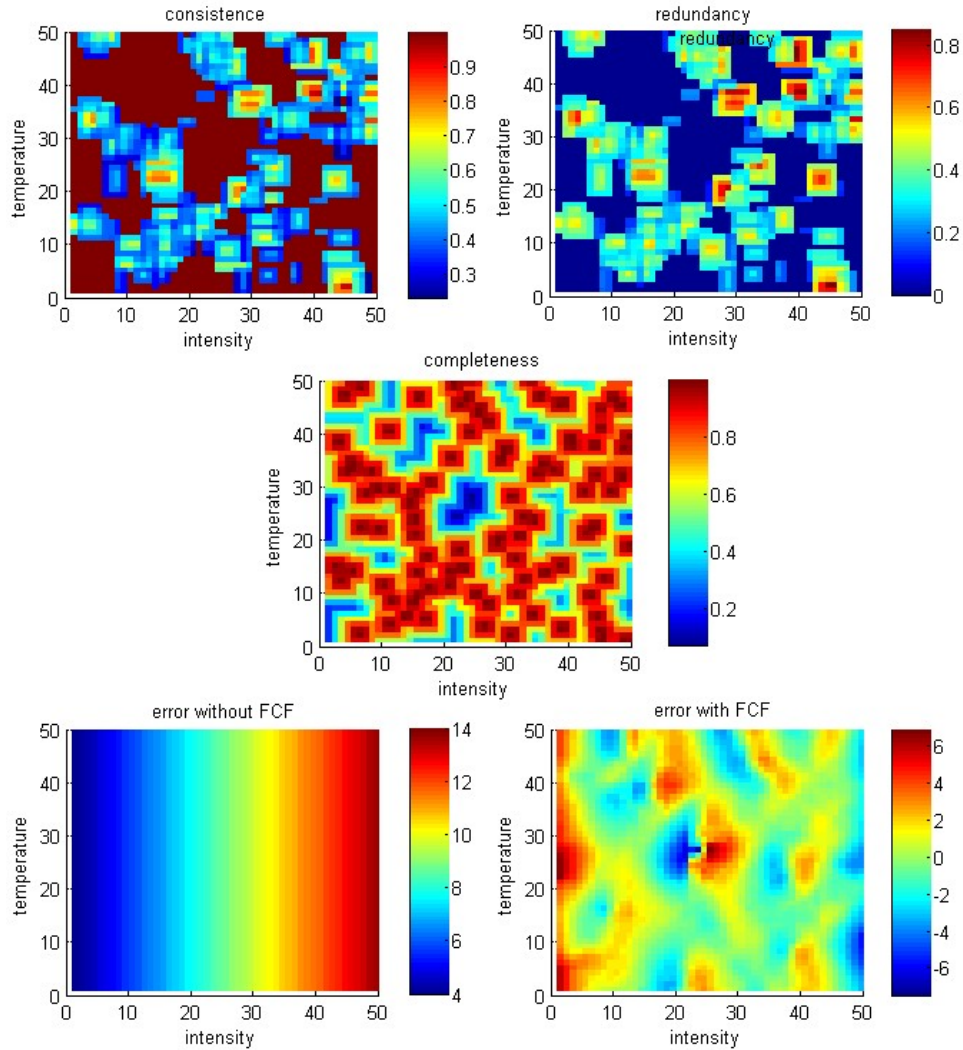


## Appendix B

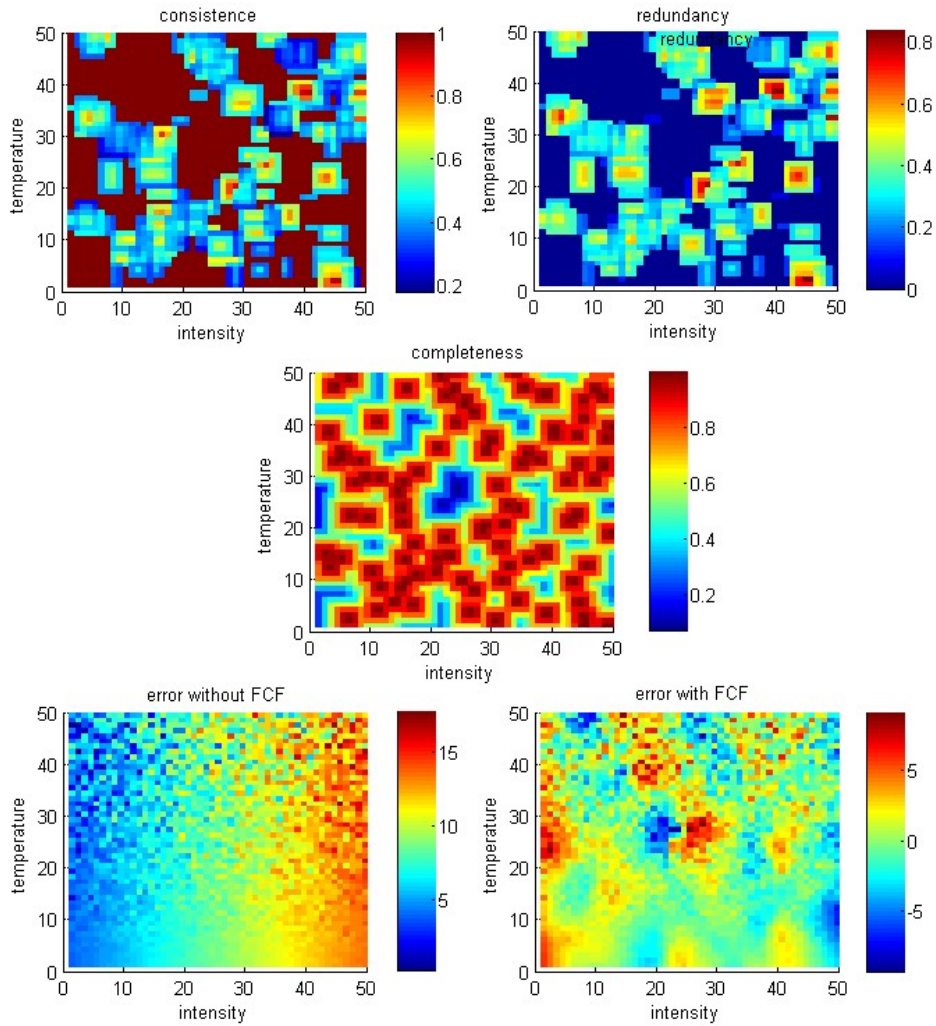
# Ammeter Calibration Case Example Results

### B.1 Ammeter Calibration Case Example Results

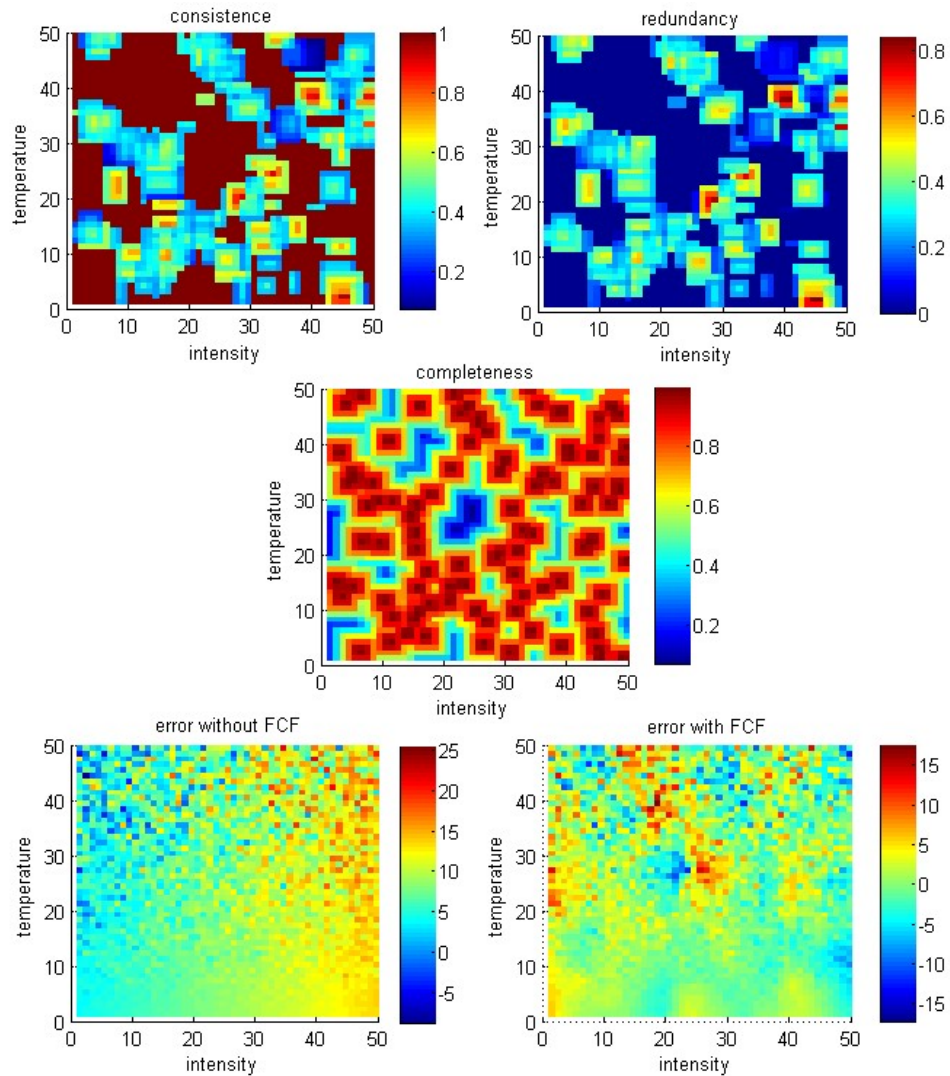
This appendix contains the results regarding local reliability criteria for the ammeter calibration case example. They complete the results given in Figures to in Chapter 2. Each figure shows detailed graphs of the local consistency, redundancy and completeness reliability criteria values jointly with graphs of the errors committed by the ammeter without using the FCF and using it for different values of the parameters  $n$  (number of rules),  $c$  (fuzzy number width) and  $K$  (random error standard deviation). Figures correspond to the following set of values for  $n$  and  $c$ :  $\{(n = 100, c = 2), (n = 100, c = 4), (n = 1000, c = 1), (n = 1000, c = 2)\}$ . For every pair, each value of  $K$  in  $\{0, 0.01, 0.05, 0.1, 0.2, 0.5\}$  is analyzed.



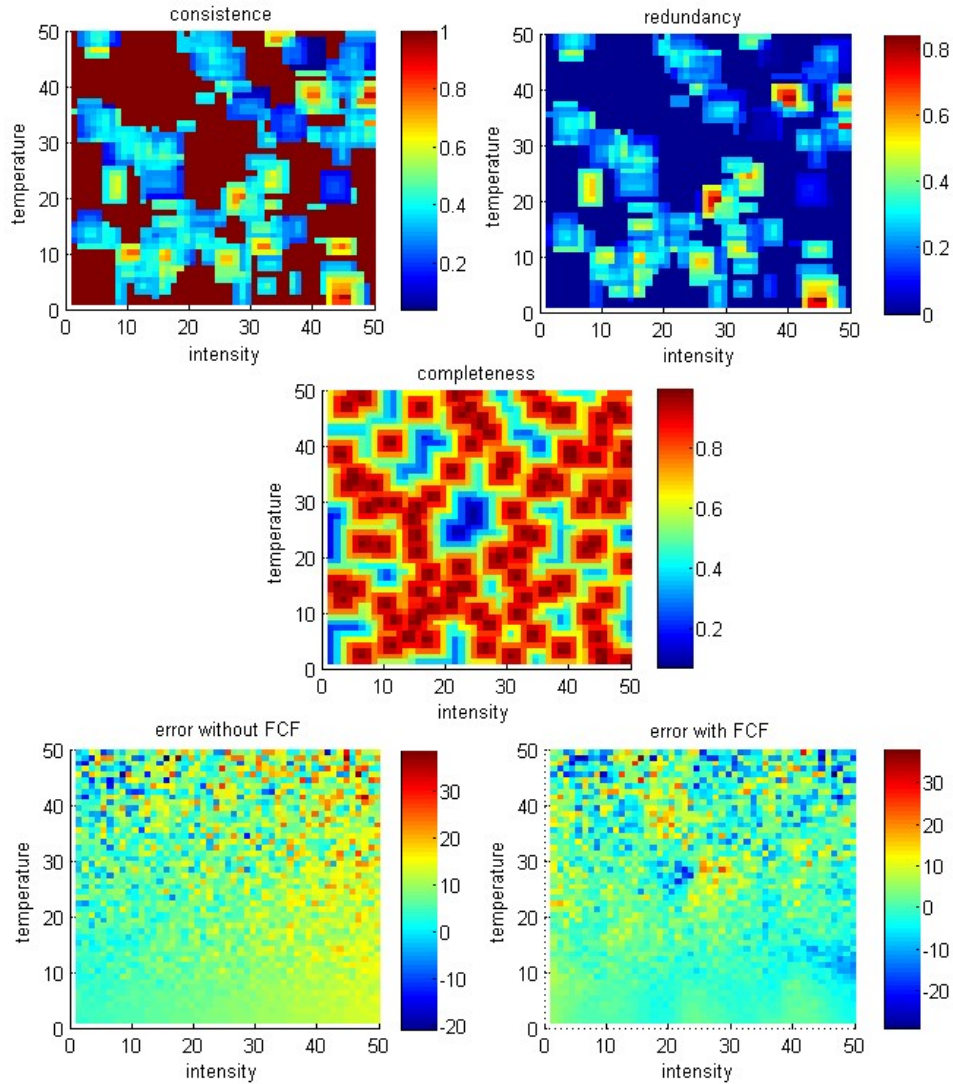
**Figure B.1:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 4$ , random error standard deviation  $K = 0$ .



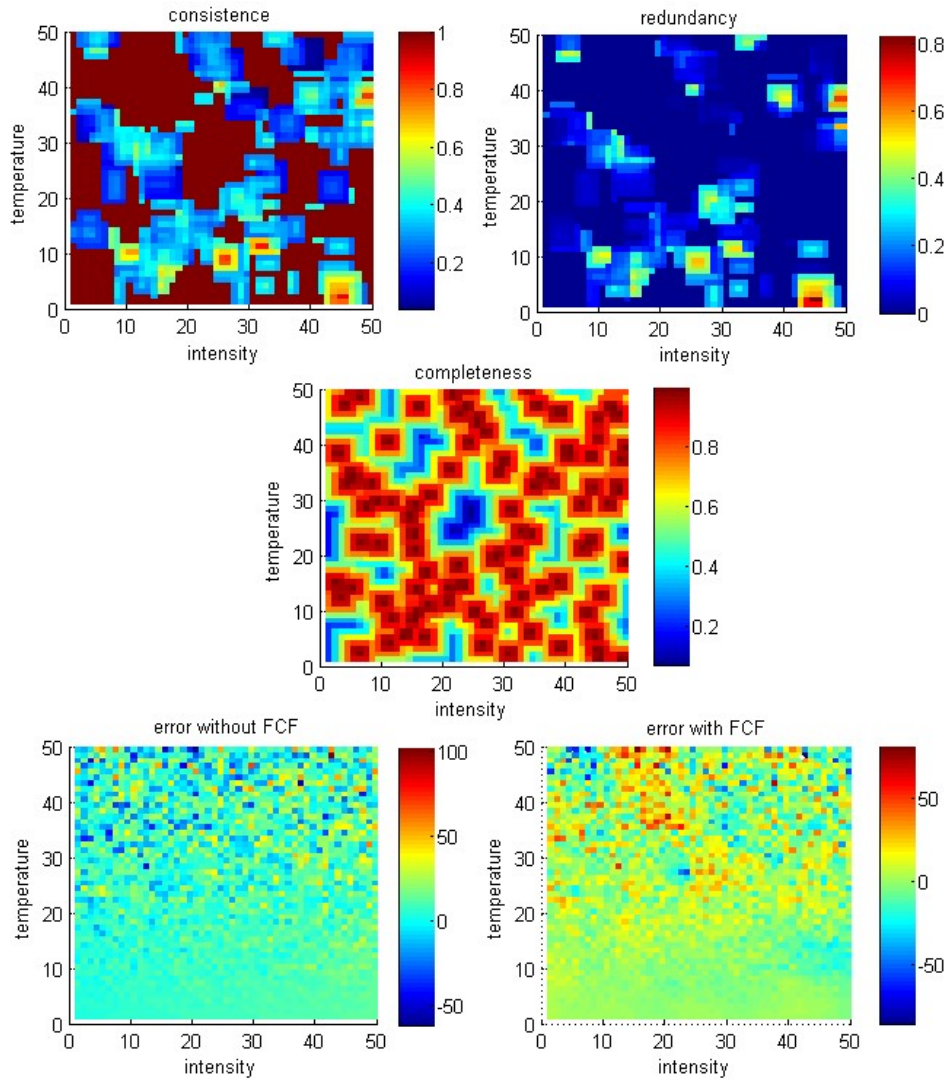
**Figure B.2:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 4$ , random error standard deviation  $K = 0.05$ .



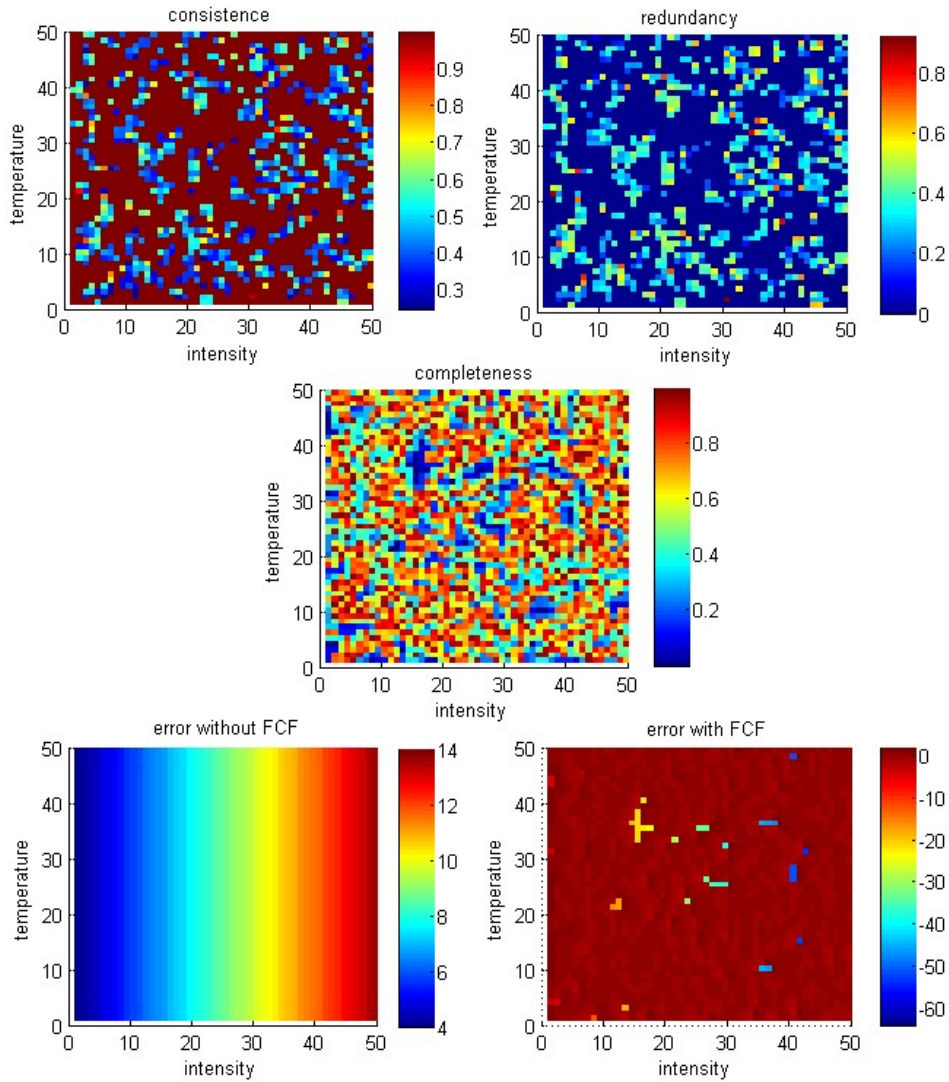
**Figure B.3:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 4$ , random error standard deviation  $K = 0.1$ .



**Figure B.4:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 4$ , random error standard deviation  $K = 0.2$ .

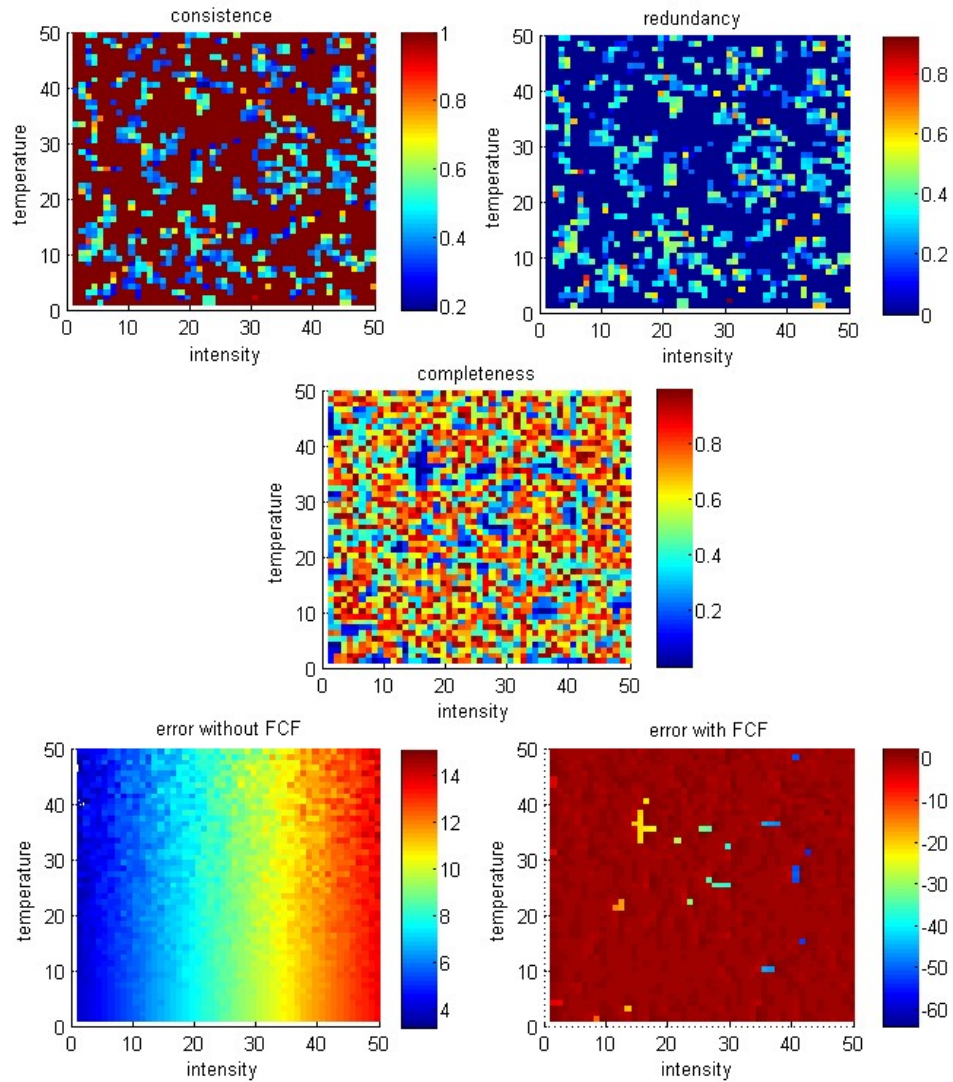


**Figure B.5:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 4$ , random error standard deviation  $K = 0.5$ .

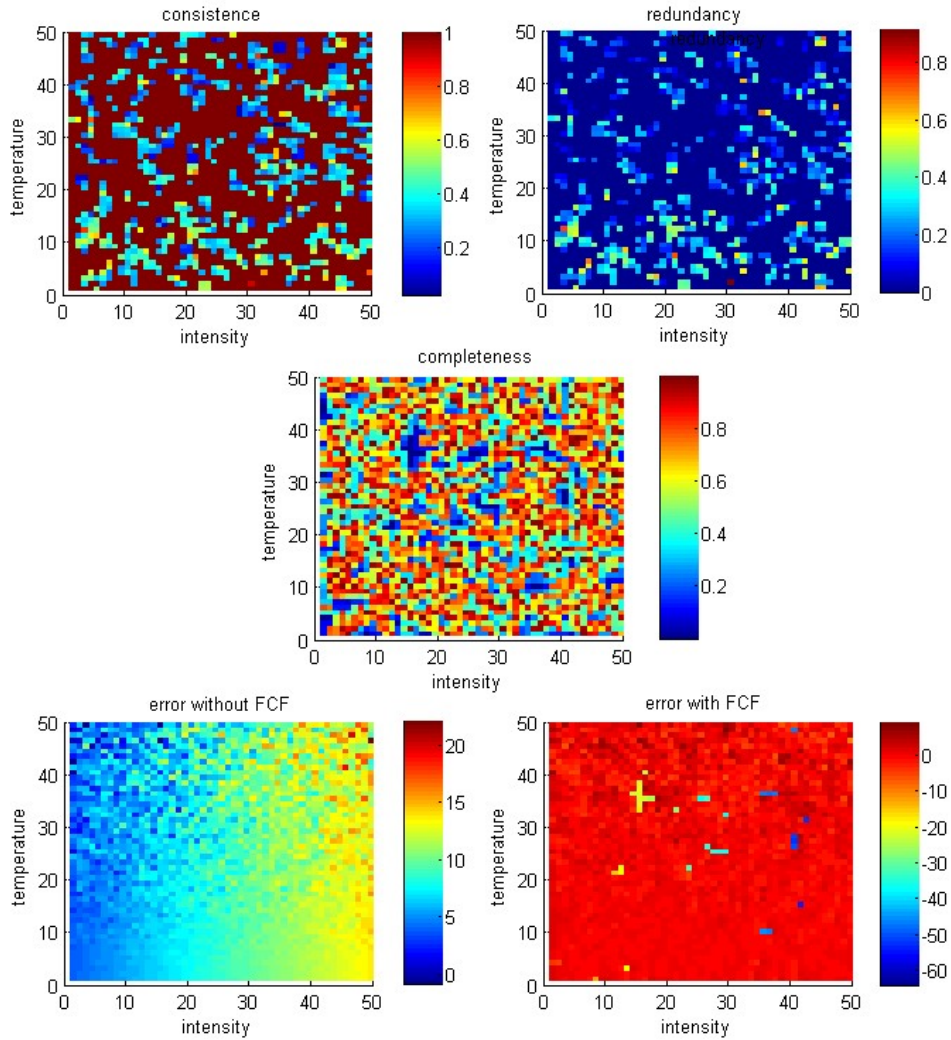


**Figure B.6:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 1$ , random error standard deviation  $K = 0$ .

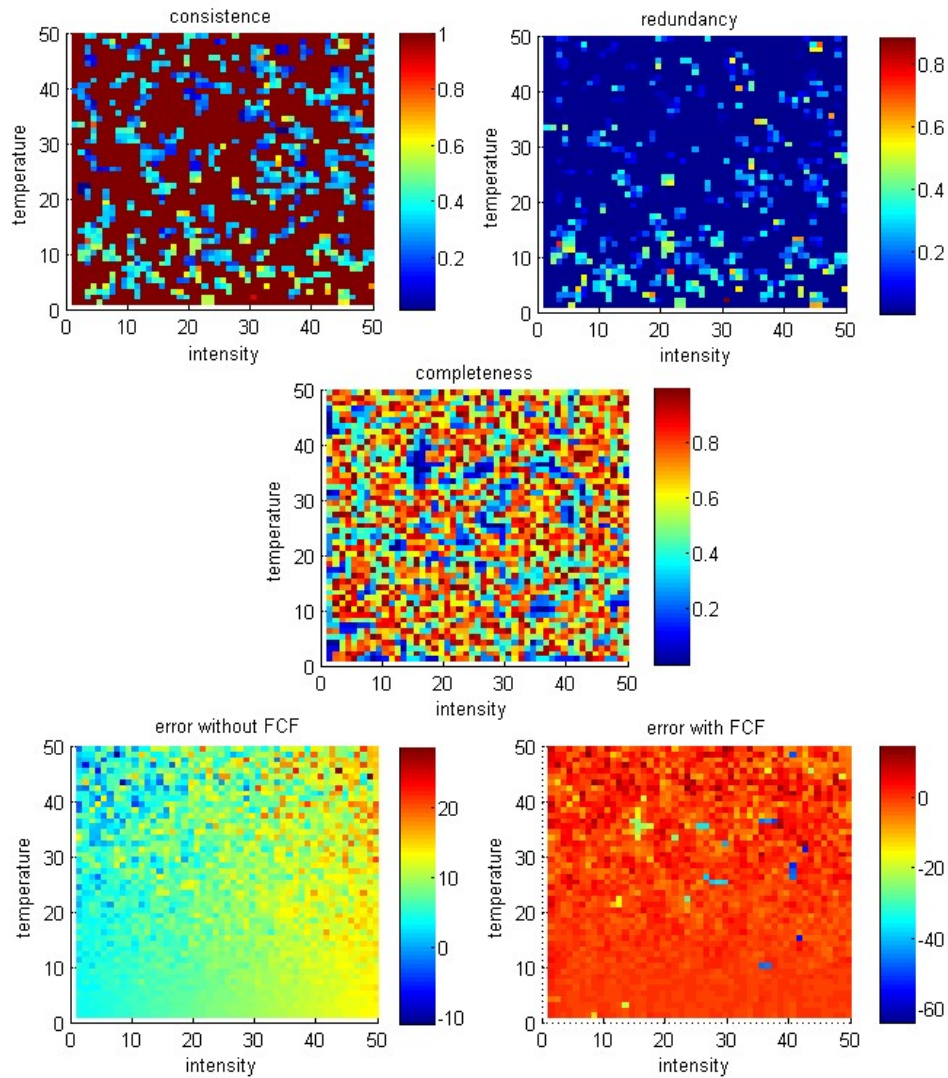




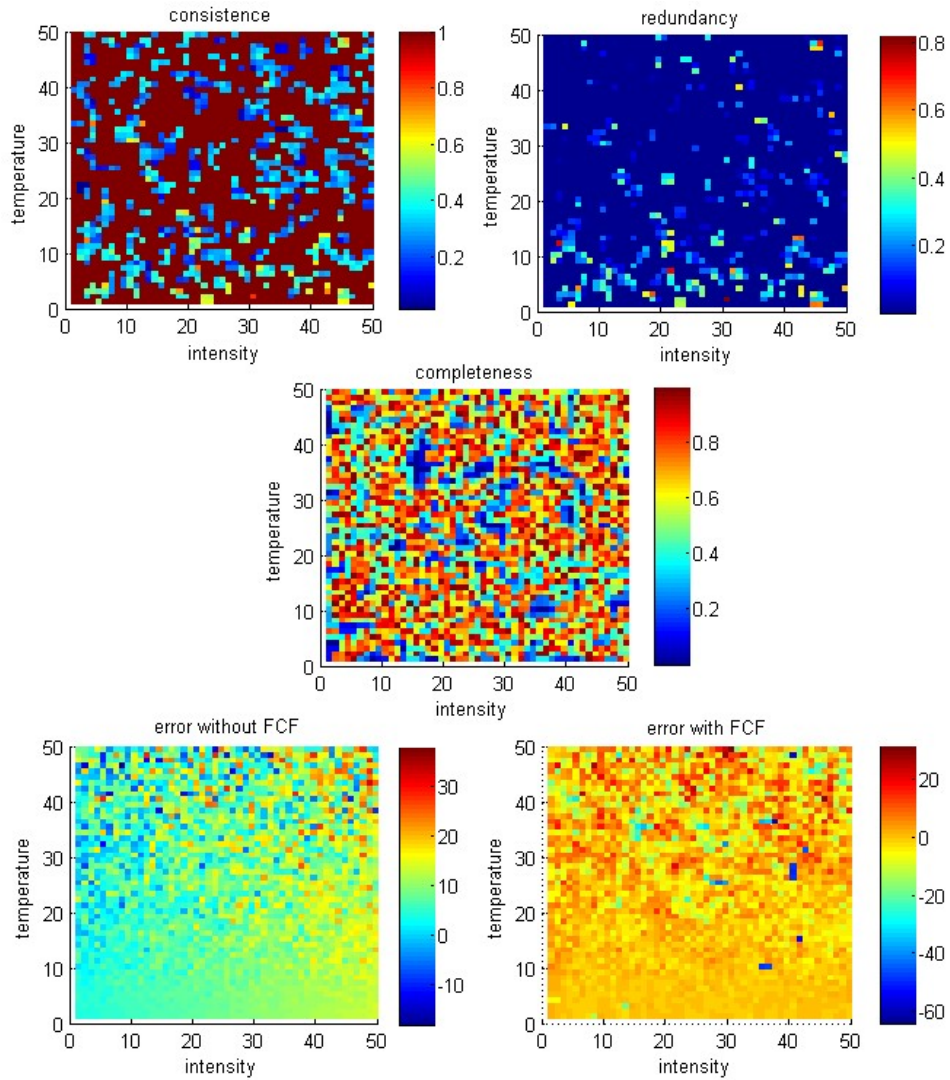
**Figure B.7:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 1$ , random error standard deviation  $K = 0.01$ .



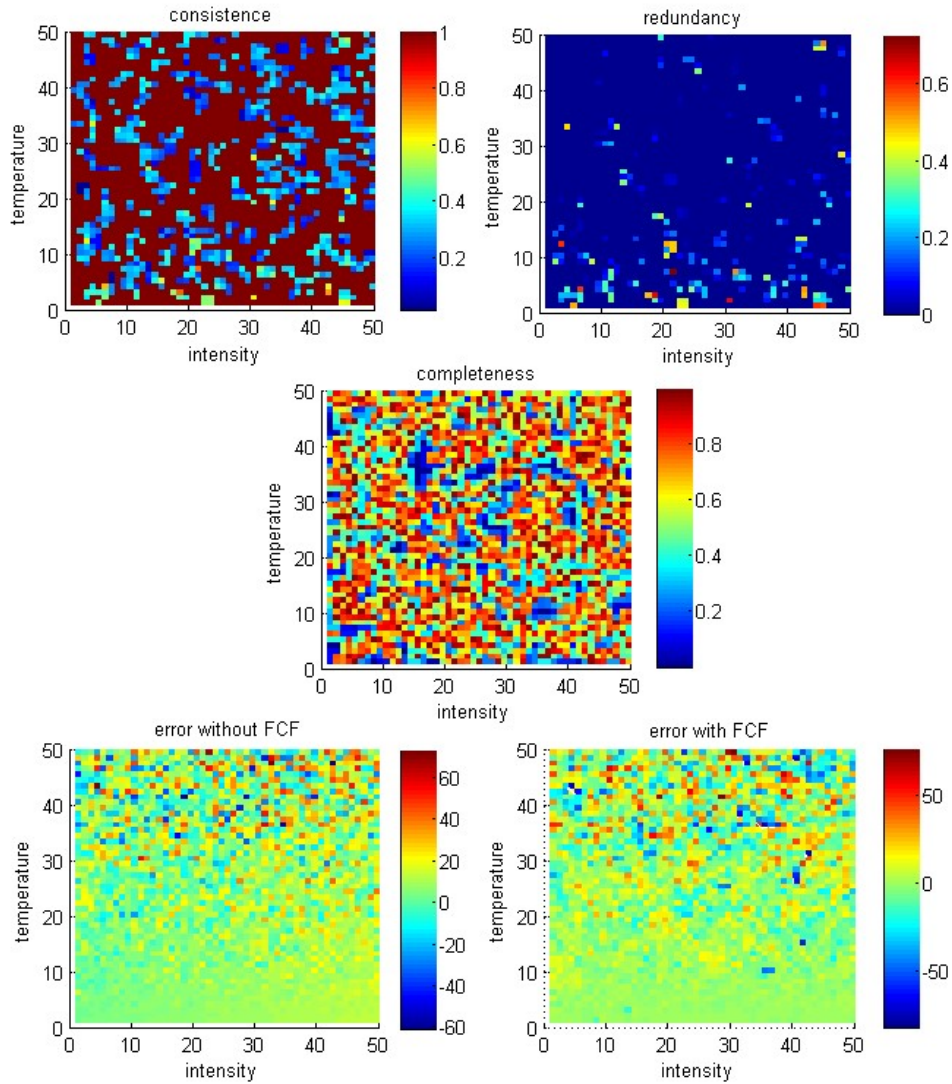
**Figure B.8:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 1$ , random error standard deviation  $K = 0.05$ .



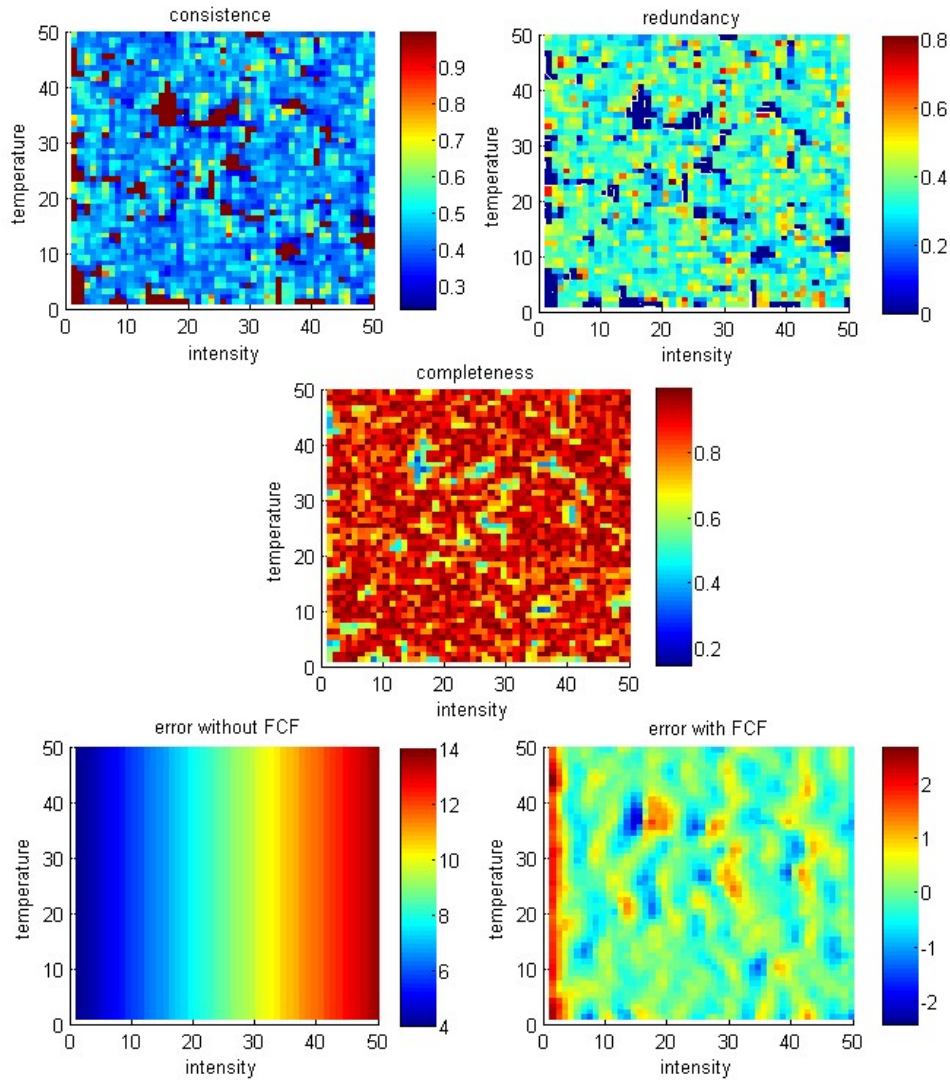
**Figure B.9:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 1$ , random error standard deviation  $K = 0.1$ .



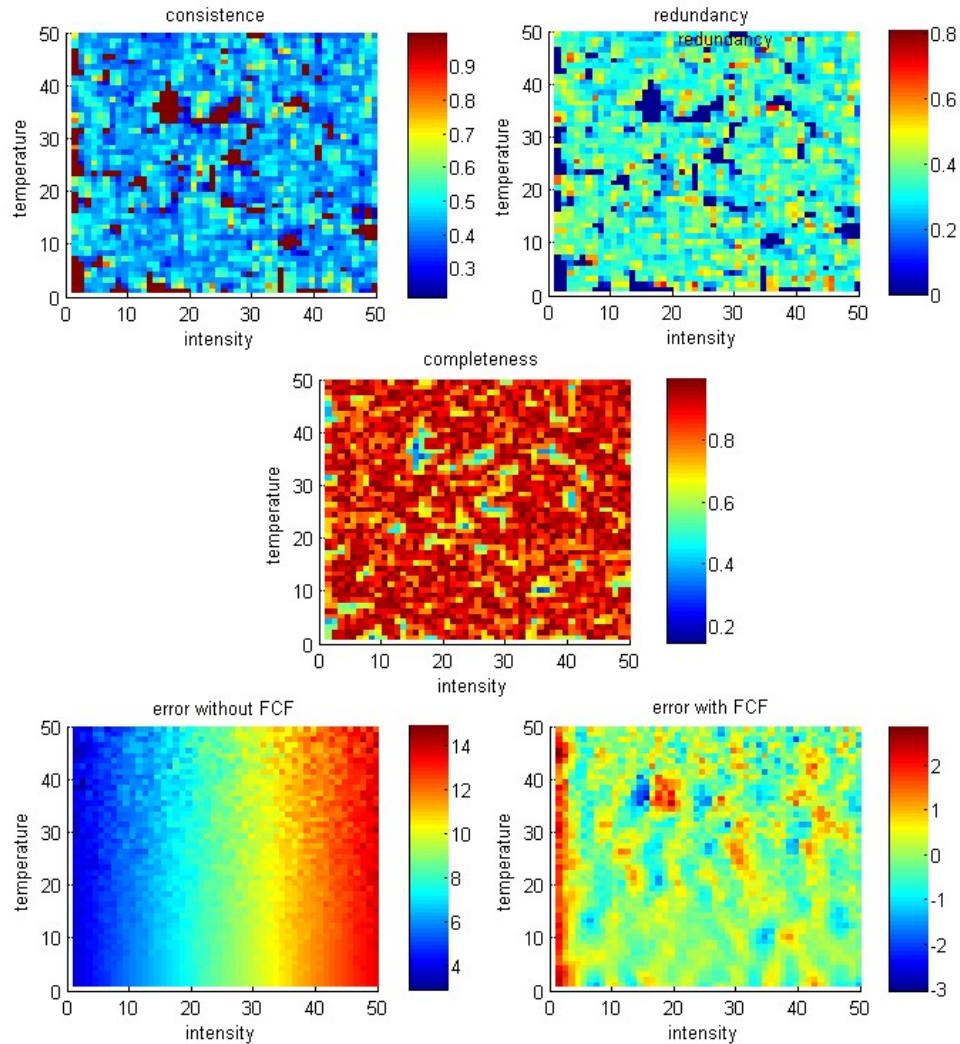
**Figure B.10:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 1$ , random error standard deviation  $K = 0.2$ .



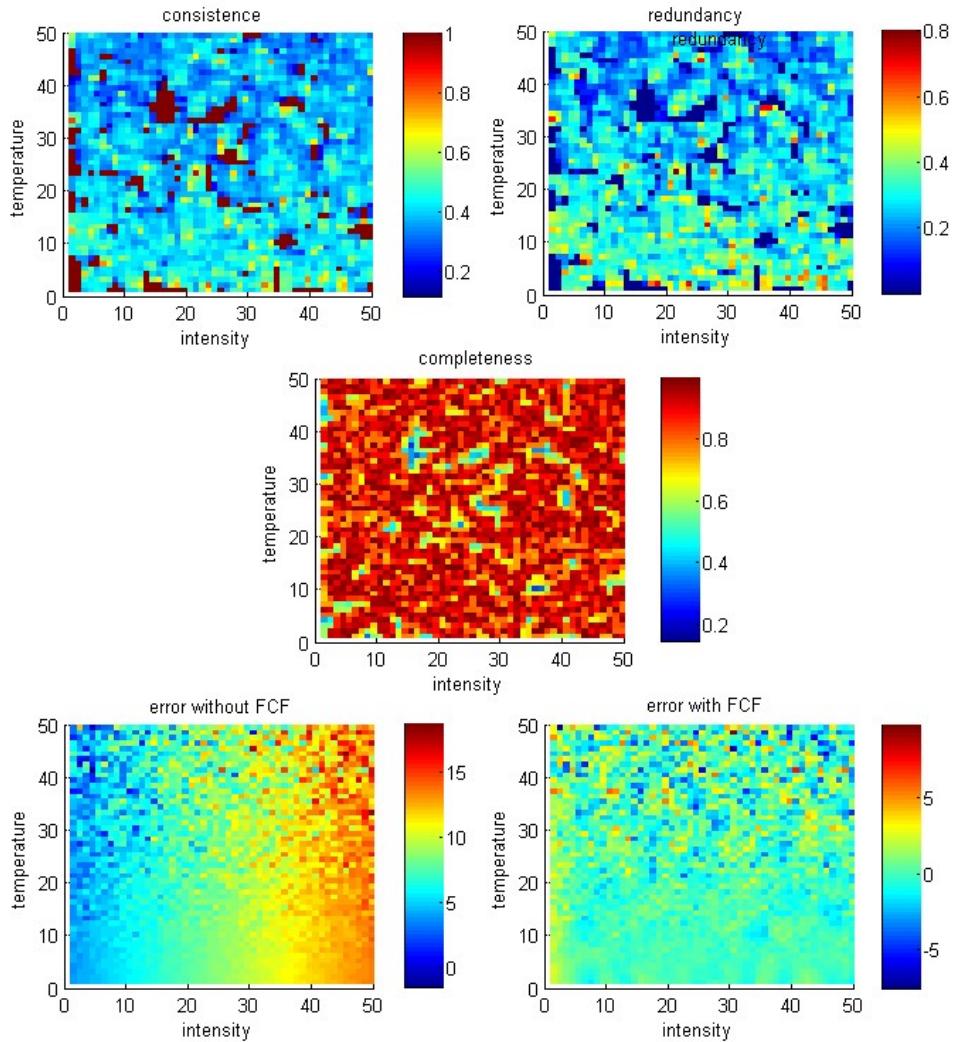
**Figure B.11:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 1$ , random error standard deviation  $K = 0.5$ .



**Figure B.12:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0$ .

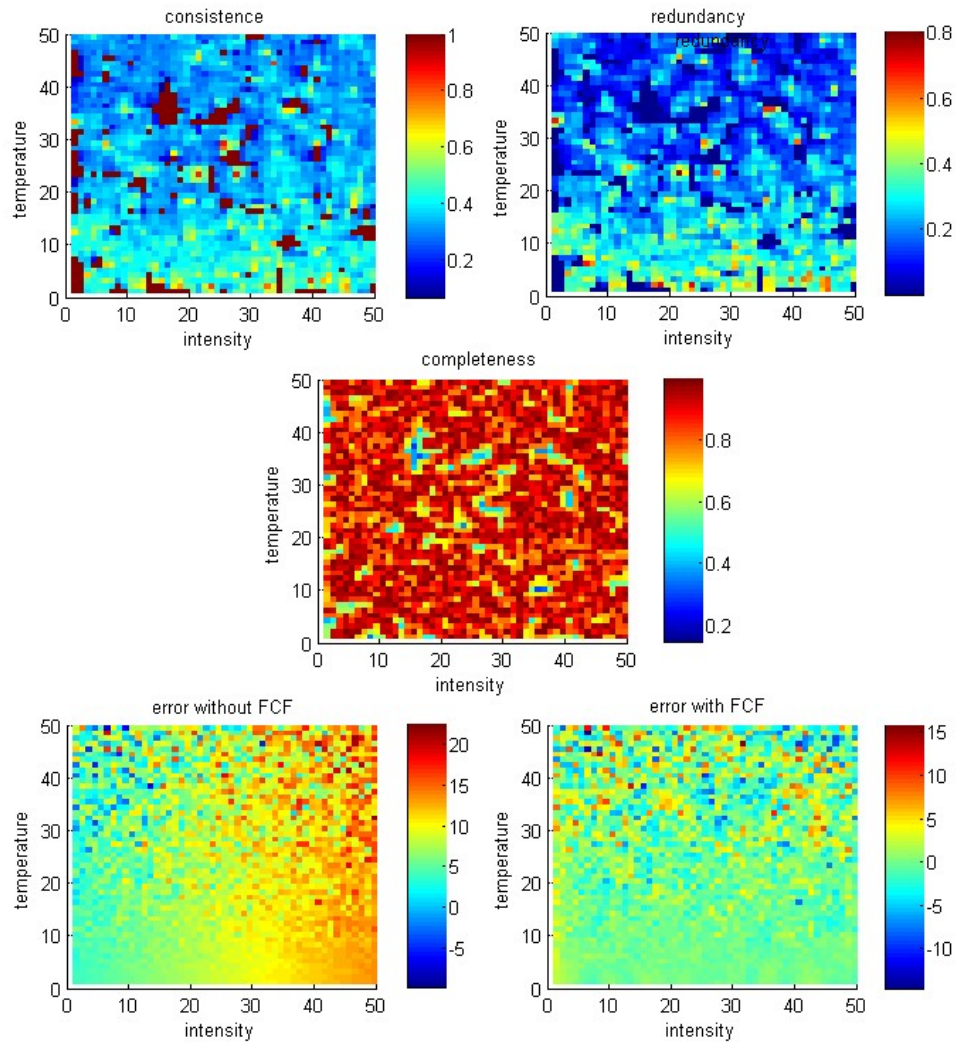


**Figure B.13:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0.01$ .

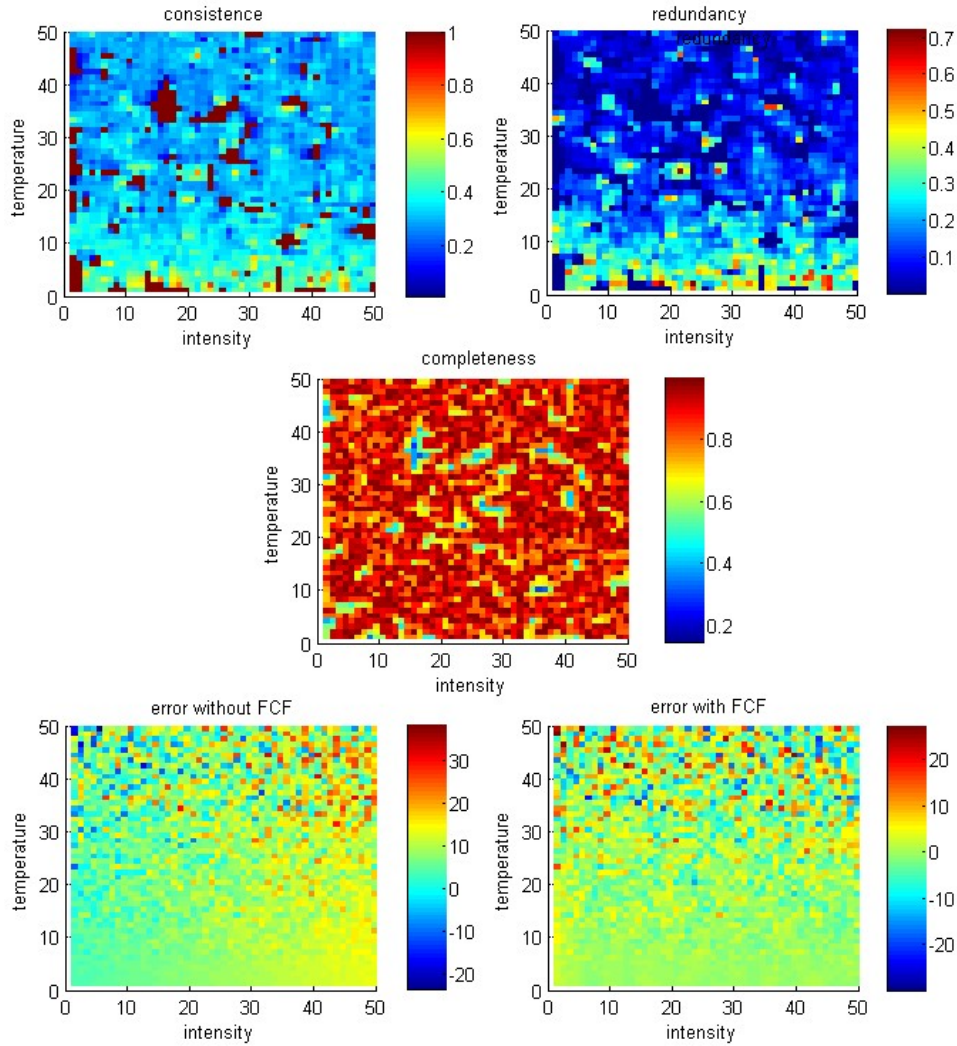


**Figure B.14:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0.05$ .

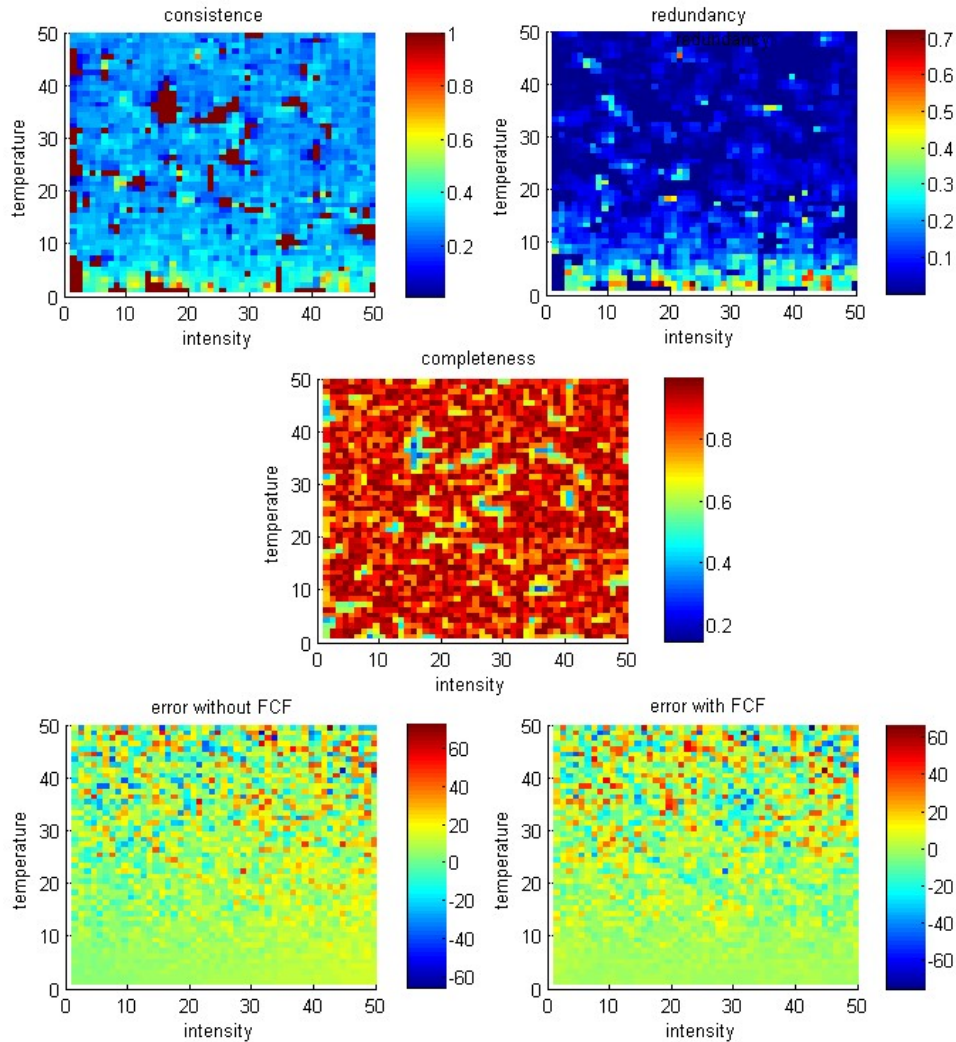




**Figure B.15:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0.1$ .



**Figure B.16:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 1000$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0.2$ .



**Figure B.17:** Top and middle rows: detailed graphs of the local consistence, redundancy and completeness reliability criteria values. Bottom row: graphs of the errors committed by the ammeter without using the FCF (to the left) and using it (to the right). Number of rules  $n = 100$ , fuzzy number width  $c = 2$ , random error standard deviation  $K = 0.5$ .

## Appendix C

# The Half Normal Distribution

### C.1 The Half Normal Distribution

The Half-normal distribution (see Figure C.1), a particular case of the Folded Normal distribution, is the probability distribution of the absolute value of a random variable normally distributed with expected value 0. That is, if  $X$  is a random variable normally distributed with mean 0, then  $Y = |X|$  is half-normally distributed.

The probability distribution function for a Half-normal distribution is:

$$\begin{aligned} P(x) &= \frac{2}{\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{x^2}{2\sigma^2}\right), \text{ if } x \geq 0 \\ &= 0, \text{ if } x < 0 \end{aligned}$$

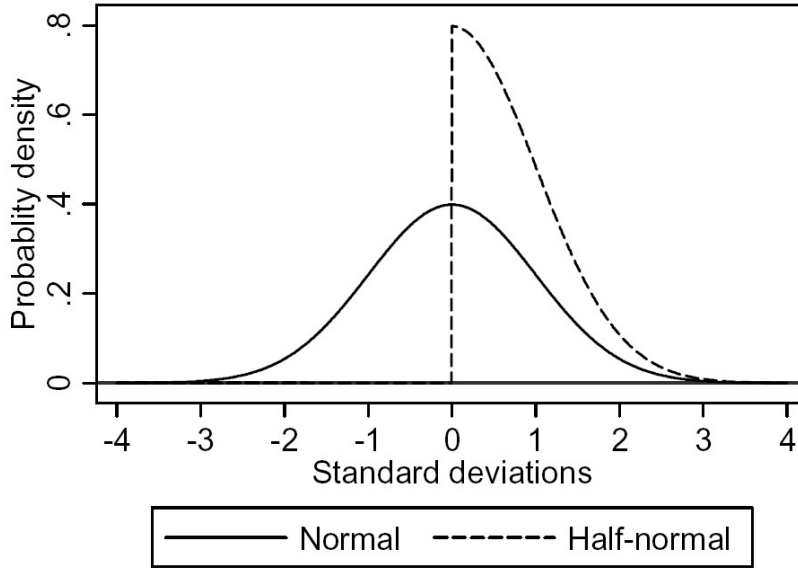
and its expected value is given by

$$E(x) = \frac{2}{\sigma\sqrt{2\pi}} \int_0^{\infty} x \cdot \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \quad (\text{C.1})$$

This is easy to integrate. We have that

$$\frac{d}{dx} \exp\left(-\frac{x^2}{2\sigma^2}\right) = -\frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

So



**Figure C.1:** Probability distribution functions of the Normal distribution with mean zero and the associated Half-Normal distribution.

$$\begin{aligned}
 E(x) &= \frac{2}{\sigma\sqrt{2\pi}} \int_0^{\infty} x \cdot \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \\
 &= \frac{2}{\sigma\sqrt{2\pi}} \int_0^{\infty} (-\sigma^2) \cdot \frac{d}{dx} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx \\
 &= \frac{-2\sigma^2}{\sigma\sqrt{2\pi}} \left[ \exp\left(-\frac{x^2}{2\sigma^2}\right) \right]_0^{\infty} \\
 &= -\sqrt{\frac{2}{\pi}} \cdot \sigma \cdot [0 - 1] = \sqrt{\frac{2}{\pi}} \cdot \sigma
 \end{aligned} \tag{C.2}$$

We have, then, the following result: If  $X$  is a random variable normally distributed with mean 0 and standard deviation  $\sigma$ , then  $Y = |X|$  follows a Half-normal distribution with mean  $\sqrt{\frac{2}{\pi}} \cdot \sigma$ .

In the ART testbed [32], an appraiser's expertise, defined as its ability to generate an opinion about the value of a painting, is described by a normal distribution of the error between the appraiser's opinion and the true painting value. The simulation creates opinions according to this error distribution, which has a mean of zero and a standard deviation  $s$  given by

$$s = \left( s^* + \frac{\alpha}{c_g} \right) t \tag{C.3}$$

where  $s^*$ , unique for each era, is assigned to an appraiser from a uniform distribution in  $[0 \dots 1]$ ,  $t$  is the true value of the painting to be appraised,  $\alpha$  is a parameter, chosen by the experimenter and fixed for all appraisers, which affects the relationship between opinion-generation cost and resulting accuracy and  $c_g$  is the cost the appraiser is willing to pay to generate an opinion.

So, what is the expected error in absolute value in a single appraisal?. The appraisal error is normally distributed with mean 0 and standard deviation given by Equation C.3. So, by the result obtained in Equation C.2, the expected value of the absolute value of the error will in a single appraisal will be  $\sqrt{\frac{2}{\pi}} \cdot \left(s^* + \frac{\alpha}{c_g}\right) t$ . Conversely, the relative error (that is, the appraisal error divided by the true value of the painting) is also a normal variable with mean 0 and standard deviation  $\left(s^* + \frac{\alpha}{c_g}\right)$ , so the expected value of the absolute value of the relative error will be  $\sqrt{\frac{2}{\pi}} \cdot \left(s^* + \frac{\alpha}{c_g}\right)$ .

Appraisers whose final appraisals are most accurate are rewarded with a larger share of the client base in subsequent timesteps. To calculate  $a$ 's share of the client base for the next simulation, the average relative appraisal error for the present iteration,  $\epsilon_a$  is first calculated:

$$\epsilon_a = \frac{\sum_{q \in T_a} \frac{|q.AppVal - q.TrueVal|}{q.TrueVal}}{|T_a|} \quad (C.4)$$

where  $T_a$  is a set of triplets in the form  $(Painting, AppVal, TrueVal)$  representing the set of appraisals made by  $a$ . For each appraisal  $q \in T_a$ ,  $q.AppVal$  and  $q.TrueVal$  represent respectively the value by which  $q.Painting$  was appraised and its true value.

We can express the expected value of  $\epsilon_a$  as:

$$E(\epsilon_a) = \frac{1}{|T_a|} \cdot \sum_{q \in T_a} E \left( \left| \frac{q.AppVal - q.TrueVal}{q.TrueVal} \right| \right) \quad (C.5)$$

But, as we have just seen,  $\left| \frac{q.AppVal - q.TrueVal}{q.TrueVal} \right|$  is the absolute value of the relative error in the appraisal of painting  $q$ , a half-normally distributed variable with expected value  $\sqrt{\frac{2}{\pi}} \cdot \left(s_q^* + \frac{\alpha}{c_g^q}\right)$ , where  $s_q^*$  is the base expertise of agent  $a$  for the era to which painting  $q$  belongs and  $c_g^q$  is the amount of money spent by agent  $a$  in the appraisal of the painting.. We can substitute those values in Equation C.5 to obtain the expected value of  $\epsilon_a$  as

$$E(\epsilon_a) = \frac{\sqrt{2/\pi}}{|T_a|} \cdot \sum_{q \in T_a} \left( s_q^* + \frac{\alpha}{c_g^q} \right) \quad (C.6)$$



## Appendix D

# The Complexity of the CONTS Problem

A les planes següents podem veure diferents instàncies del problema juntament amb les seves solucions. A la figura D.1 vegem com una petita variació a les condicions inicials fa que la solució sigui completament diferent. El problema és encara més difícil del que sembla a primera vista; per tal d'aconseguir portar tots els contenidors a temps, de vegades és necessari carregar un contenidor durant un temps i desar-lo temporalment a algun punt diferent del de lliurament per anar després a portar altres contenidors i, finalment, tornar a recollir-lo. (figura D.2, a dalt). Més encara, la figura D.2, a baix, mostra un problema pel qual la única solució passa per transportar temporalment un contenidor a un punt que ni tan sols es troba a la recta que uneix la posició inicial del contenidor amb el seu lloc de lliurament.

Aquestes solucions s'han trobat mitjançant un algorisme de cerca exhaustiva implementat en Prolog [18]. El temps de càlcul necessari per a trobar les solucions és de l'ordre de deu segons, però creix molt ràpidament amb el nombre de contenidors i robots.

### D.1 CONTS és NP-Hard

És clar que, abans d'intentar trobar mètodes per a solucions el problema, ens hem de fer una idea de la seva complexitat. Un mètode heurístic per a trobar solucions aproximades mitjançant agents reactius no tindria sentit si existissin mètodes directes de complexitat polinòmica. De la mateixa manera, si el problema és de complexitat no deterministicament polinòmica (*NP*), el més probable és que no podrem fer servir mètodes exhaustius i ens haurem de centrar en mètodes que proporcionin solucions aproximades.



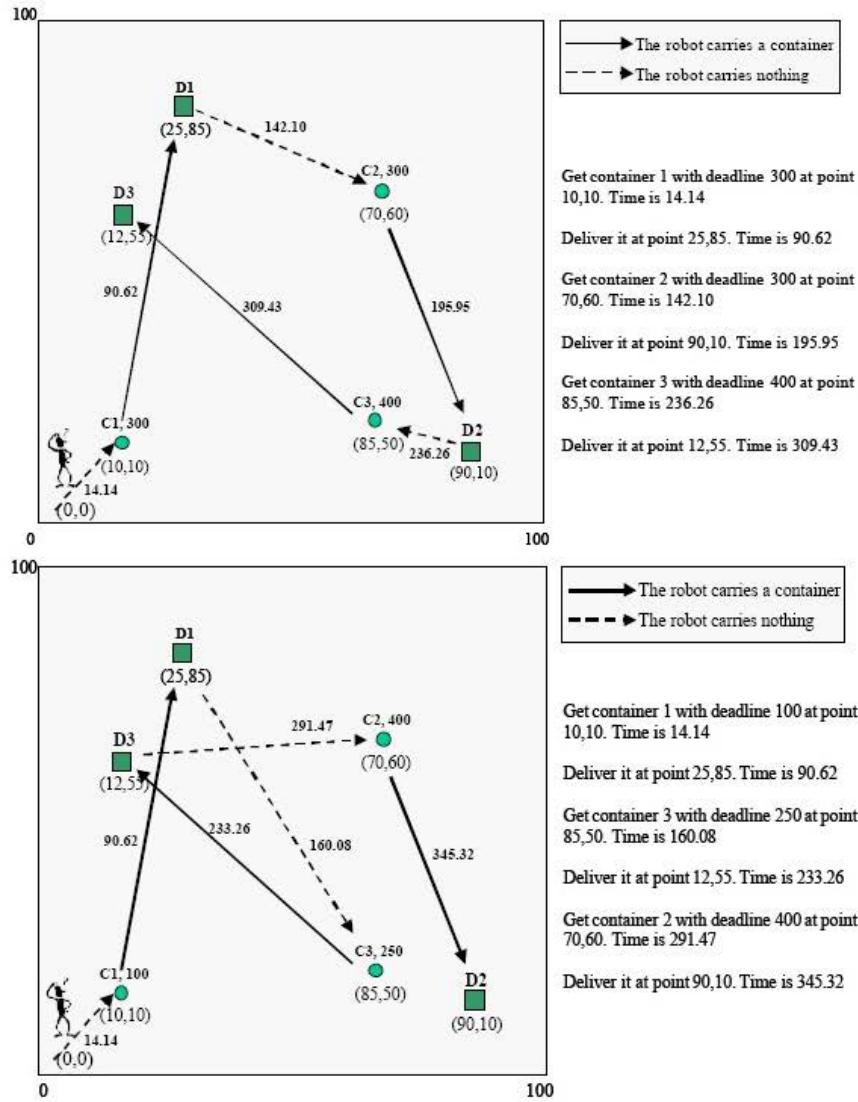
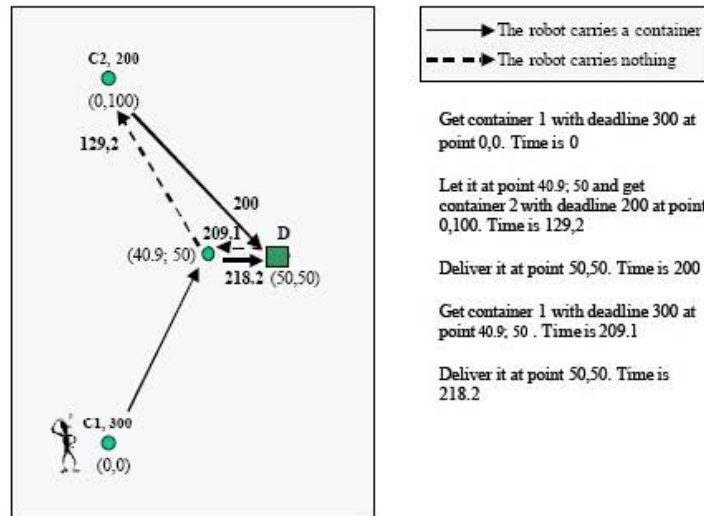
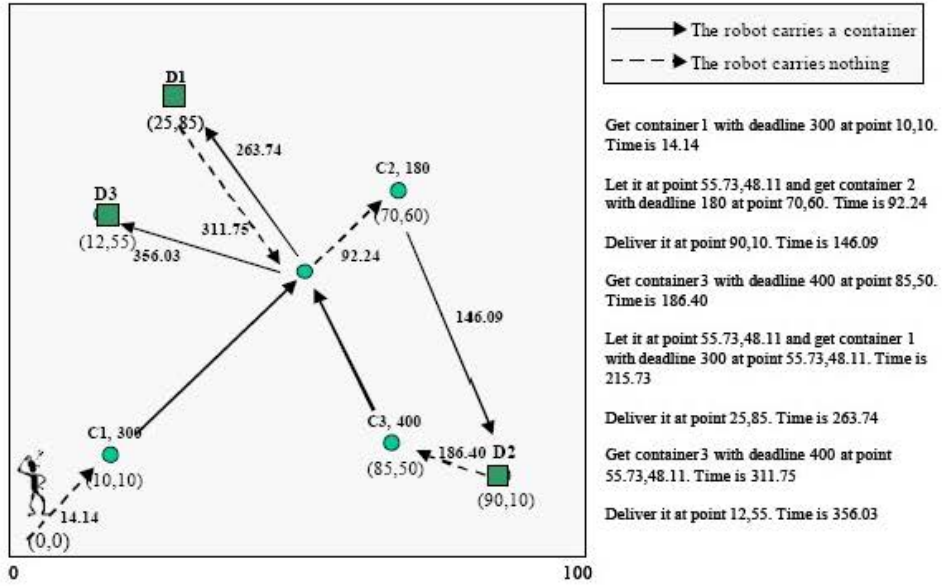


Figure D.1: Dues instàncies del problema. Els contenidors estan re-presentats amb els cercles i els llocs de lliurament amb els quadrats. A sota de cadascun dels objectes hi ha la seva posició. A sobre dels contenidors hi figura el seu deadline. Les seqüències de fletxes representen les solucions dels problemes, trobades amb un programa que implementa un mètode de cerca exhaustiu (a la dreta es pot veure la sortida del programa).



**Figure D.2:** A dalt, de vegades cal deixar un contenidor pel camí i tornar després a recollir-lo. A baix, tots dos contenidors s'han de lliurar al mateix punt. La única solució pel problema exigeix deixar el contenidor C1 a un punt exterior a les rectes que van dels contenidors al punt de lliurament.

Ja hem comentat en l'apartat anterior com el temps necessari per a fer la cerca exhaustiva dintre de l'espai de solucions del problema augmenta molt ràpidament amb el nombre de contenidors i robots. El propòsit d'aquesta secció és demostrar que el problema és prou complex com per que cap algorisme exhaustiu sigui viable, de manera que quedarà justificada la utilització d'heurístiques per a tractar de trobar solucions prou bones. Demostrarem que el problema és, al menys, tant complex com el famós problema *NP*-hard del viatjant de comerç (més conegut com el travelling salesman problem o TSP). Per tal de fer-ho, necessitem algunes definicions preliminars.

### D.1.1 Problemes de Decisió i Classes de Complexitat

Un problema de decisió es formalitza usualment com el problema de decidir si una cadena de caràcters pertany a un llenguatge formal específic. El llenguatge conté el conjunt de cadenes pel les que la resposta és "SI". Per exemple, el problema de decidir si un nombre és o no primer es pot formalitzar com el de decidir la pertinència de cadenes de bits al llenguatge format per les paraules sobre l'alfabet binari que representen un nombre primer en base dos. Si existeix un algorisme capaç de decidir correctament per a cada possible cadena d'entrada si pertany o no al llenguatge, llavors el problema es diu que és decidible; en cas contrari es diu que és indecidible. Si existeix un algorisme que sempre respon "SI" quan la cadena d'entrada pertany al llenguatge, però que s'executa indefinidament sense donar cap resposta quan la cadena no pertany al llenguatge, llavors el llenguatge es diu que és parcialment decidible. Alguns exemples de problemes de decisió expressats com a llenguatges són:

- Les cadenes sobre a,b formades per as i bes alternades.
- Les cadenes sobre a,b amb el mateix nombre d'as i bes.
- Les cadenes sobre 0,1, ..., 9 que representen un nombre primer.
- Les cadenes que descriuen un conjunt de sencers tals que algun subconjunt d'ells suma 0.
- Les cadenes que descriuen una màquina de Turing i una cinta d'entrada tal que la màquina de Turing s'atura amb aquesta cinta.

Els problemes de decisió són importants perquè qualsevol problema més general, amb resposta d' $n$  bits, es pot transformar en, com a molt,  $n$  problemes de decisió. Segons la Teoria de la Complexitat, la classe *P* consisteix en tots aquells problemes de decisió que es poden resoldre en un màquina seqüencial determinista - com per exemple un ordinador programable amb memòria no limitada o una màquina de Turing- en una quantitat de temps que creix

polinòmicament amb la longitud de la cadena d'entrada. La classe  $NP$  consisteix en aquells problemes de decisió tals que les seves solucions positives poden ser verificades, donada la informació necessària, en temps polinòmic amb la longitud de l'entrada. Equivalentment, la classe  $NP$  també es pot definir com la d'aquells problemes que es poden solucionar en temps polinòmic amb l'entrada fent servir una màquina no determinista (per tant,  $NP$  prové de Non-deterministic Polynomial-time, i no de Non-Polynomial time).

Com tothom sap, l'interrogant no resolt més important que actualment es planteja la Informàtica Teòrica és si ambdues classes són la mateixa. En essència, la pregunta que la coneguda equació  $P = NP$  planteja és la següent: si les solucions a un problema de decisió poden ser verificades ràpidament (en temps polinòmic), això implica que puguin ser calculades ràpidament?. Pensem en el problema de determinar si un nombre (per exemple el 69779) és primer. Resulta que el 69779 és un nombre compost, encara que és força costós esbrinar-ho. Per altra part, si algú ens diu que el 69779 no és primer perquè és divisible per 223, només necessitem fer una divisió per verificar-ho. En el nostre problema, donada la informació correcta (223 divideix a 69779), verificar la resposta (69779 no és primer) és ràpid, ja que es pot fer en temps polinòmic amb la longitud de l'entrada. Per tant el problema és a  $NP$ . La qüestió de si el problema pertany a  $P$  no ha estat clara durant segles, fins que s'ha resolt definitivament fa poc temps [2]. en que s'ha trobat un algorisme de complexitat  $O(n^{12})$  que el soluciona, on  $n$  és la longitud del nombre. (que no és el mateix que la seva magnitud).

No tots els problemes són a  $NP$ . De fet hi ha problemes que són més difícils que qualsevol problema a  $P$  o a  $NP$  (per exemple problemes relacionats amb trobar estratègies guanyadores a jocs com els escacs o el go [30, 5]). No obstant, dintre de  $NP$  hi ha una subclasse de problemes, els  $NP$ -complets, interessants per diferents conceptes. Per una part perquè no s'ha trobat cap algorisme de temps polinòmic que els resolgui, i per l'altra perquè s'ha demostrat que tot problema a  $NP$  es pot reduir a un d'ells, de manera que un algorisme que permetés resoldre qualsevol problema  $NP$ -complet en temps polinòmic es podria fer servir per a solucionar-los tots també en temps polinòmic. Hi ha molts de problemes  $NP$ -complets molt coneguts i de gran rellevància pràctica, com són, per exemple, el problema de la satisfacibilitat booleana, el problema de la motxilla, el problema del viatjant de comerç, el 15-puzzle i, fins i tot, el popular buscamines.

Una altra classe de complexitat molt relacionada amb les anteriors és la dels problemes  $NP$ -hard. Una definició intuïtiva seria "tant complex o més que un problema  $NP$ -complet". La classe  $NP$ -hard no es limita als problemes de decisió, sinó que també inclou altres tipus de problemes. Per exemple, el problema  $NP$ -complet "donat un conjunt de sencers, dir si conté cap subconjunt que sumi zero" té el problema  $NP$ -hard associat "donat un

conjunt de sencers, retornar un subconjunt que sumi zero, si existeix, i el conjunt buit en cas contrari". En general tots els problemes *NP*-complets tenen el seu problema *NP*-hard associat, i es demostra fàcilment que la complexitat de tots dos problemes és la mateixa. És a dir, donada una caixa negra (també anomenada oracle) que resolgui un d'ells en una unitat de temps, és possible trobar un algorisme que resolgui l'altre en temps polinòmic amb la longitud de l'entrada. També existeixen problemes de decisió *NP*-hard que no son *NP*-complets (de fet *NP*-complet és la intersecció de *NP* i *NP*-hard), com el problema de l'aturada. Per als lectors interessats en aprofundir en matèries com les diferents classes de complexitat i els problemes que pertanyen a elles, [34] és la referència clàssica a consultar. El fet que no es conegui cap algorisme eficient per als problemes *NP*-complets i els seus problemes *NP*-hard associats fa que de vegades per a resoldre'ls calgui fer servir mètodes heurístics. (és a dir, mètodes de complexitat polinòmica que no garanteixen trobar solucions exactes). En el nostre cas, per tal de justificar plenament l'ús d'un sistema d'agents reactius per a resoldre el problema CONTS dels contenidor i els robots, hem de demostrar que el problema és prou difícil com per a que, versemblantment, no existeixi cap solució exacta de complexitat polinòmica. Això ho farem demostrant que el problema és, al menys, tant complex com un problema *NP*-hard.

### D.1.2 El Problema del Viatjant de Comerç (TSP)

És potser el problema *NP*-hard més famós. L'enunciat és molt simple: donat un conjunt de ciutats i coneixent la distància que separa cada parell de ciutats, s'ha de trobar el circuit que minimitza la distància total recorreguda per un viatjant de comerç que ha de visitar totes i cadascuna de les ciutats abans de tornar a la ciutat d'origen. El problema sovint es formalitza fent servir teoria de grafs de la següent manera: Sigui  $G(V, A)$  un graf dirigit amb un conjunt  $V = \{v_1 \dots v_n\}$  de vèrtex que representen les ciutats i un conjunt  $A = \{a_1 \dots a_m\}$  d'arestes de la forma  $a = (v_i, v_j)$  que representen les carreteres que uneixen les ciutats. Sigui  $c(v_i, v_j)$  el cost (és a dir, la distància) associat a l'aresta  $(v_i, v_j)$ . El problema TSP consisteix en trobar el circuit hamiltonià de cost mínim a  $G$ .

Diguem DTSP al problema de decisió associat al TSP, és a dir, donats el graf  $G(V, A)$  i un cost màxim  $C$ , determinar si existeix cap circuit hamiltonià de cost més petit o igual a  $C$ . A [34] es demostra que DTSP és *NP*-complet.

Per altra part, definim un nou problema de decisió, relacionat amb el DTSP, però una mica diferent. Suposem que el viatjant no ha de tornar a casa per dormir, sinó que es queda a dormir a l'última ciutat visitada. Llavors el problema es pot reformular així: donat el graf  $G$  definit anteriorment, un cost  $C$  i un vèrtex inicial  $v_i$ , determinar si existeix un camí a  $G$  (és a dir, una seqüència de vèrtex connectats per arestes) amb cost més petit o igual

que  $C$  que, començant a  $v_i$ , passi per tots els vèrtex del graf  $G$ . Diguem-li a aquests nou problema DTSP\*.

A partir d'aquí farem el següent: demostrarem que, per una banda, és possible reduir el DTSP al DTSP\*. I que, per l'altra, el DTSP\* es pot reduir al DCONTS (el problema de decisió associat a CONTS). D'aquesta manera demostrarem que DCONTS és NP-complert i, com a conseqüència immediata, que CONTS és NP-hard.

### D.1.3 DTSP\* és NP-complert

Demostrarem que DTSP\* és NP-complert trobant una manera de reduir el DTSP al DTSP\*. Recordem que per a reduir un cert problema A a un altre problema B assumim l'existència d'una caixa negra o oracle que resol el problema B en una unitat de temps. Si podem trobar un algorisme que resolgui el problema A, fent servir l'oracle per a B, en temps polinòmic amb la longitud de l'entrada llavors direm que hem reduït el problema A al B. Llavors, per definició, si el problema A era NP-complert i el problema B era NP haurem demostrat que B també és NP-complert.

La reducció és força senzilla. Suposem que tenim un oracle per a resoldre el DTSP\*, és a dir, una funció  $dtsp^*(G(V, A), v_i, C)$  que, donat un graf  $G$  (amb un conjunt de vèrtex  $V$  i un conjunt d'arestes amb costos associats  $A$ ), un cost màxim  $C$  i un vèrtex inicial  $v_i$  ens diu en una unitat de temps si existeix cap camí que comenci a  $v_i$  i visiti tots els vèrtex amb un cost total més petit o igual que  $C$ . Llavors, podem definir una funció  $dtsp(G(V, A), C)$  que donat un graf  $G$  i un cost màxim  $C$  ens digui si hi ha cap circuit hamiltonià a  $G$  de cost menor o igual a  $C$  de la manera que s'exposa a l'Algorisme D.1:

La idea bàsica que fa servir aquesta funció és la següent: si modifiquem el graf afegint un nou vèrtex  $p$  que estigui a distància infinita de tots els altres a excepció d'un vèrtex  $w$  (que no sigui l'inicial), del qual està a distància zero, llavors és segur que al graf modificat, el camí més curt que comença al vèrtex inicial i visita tots els vèrtex acaba amb l'arc  $(w, p)$  i que la part d'aquest camí que porta del vèrtex inicial a  $w$  és el camí més curt que uneix aquests dos vèrtex visitant tots els altres. Per altra part, és segur que al circuit hamiltonià de cost mínim a  $G(V, A)$ , el camí que porta del vèrtex inicial al vèrtex final (la darrera ciutat visitada abans de tornar a la ciutat d'origen) és el camí de cost mínim que uneix els dos vèrtex i que visita totes les ciutats, i aquest camí el podem obtenir iterant la  $w$  del procediment explicat anteriorment sobre el conjunt de vèrtex  $V$ .

Està clar que la funció  $dtsp$  crida a la funció  $dtsp^*$  un màxim de  $|V| - 1$  cops i, per altra part, sabem que  $|V|$  és de l'ordre de l'arrel quadrada de la longitud de l'entrada (que és bàsicament la descripció d'un graf). Per tant, segons el que hem exposat abans, DTSP\* és NP-complert.

---

**Algorithm D.1** La funció  $dtsp(G(V, A), C)$  diu si hi ha cap circuit hamiltonià al graf  $G(V, A)$  amb un cost menor o igual a  $C$

---

```

1: function DTSP( $G(V, A)$ : graf,  $C$ : real) : boolean
2:   Sigui  $v$  un vèrtex qualsevol de  $V$ 
3:   for all  $w \in V, w \neq v$  do
4:      $newC \leftarrow C - cost(w, v)$ 
5:     Creem un nou vèrtex  $p$ 
6:      $newV \leftarrow V \cup \{p\}$ 
7:      $newA \leftarrow A \cup \{(q, p) \forall q \in V\}$ 
8:      $cost(w, p) \leftarrow 0$ 
9:      $cost(q, p) \leftarrow \infty \forall q \in V, q \neq w$ 
10:    if  $dtsp^*(G(newV, newA), v, newC)$  then
11:      return CERT
12:    end if
13:  end for
14:  return FALS
15: end function

```

---

#### D.1.4 DCONTS és NP-complert

Sigui DCONTS el problema de decisió associat a CONTS: donats un conjunt de contenidors amb deadlines associats a cadascun d'ells i un conjunt de robots dir si existeix alguna manera de poder lliurar tots els contenidors a temps. És clar que DCONTS és a *NP*. Demostrarem que és *NP*-complert reduint el DTSP\* a ell.

En efecte, qualsevol instància del problema DTSP\* amb un graf  $G(V, A)$ , un vèrtex inicial  $v$  i un cost màxim  $C$  es pot transformar immediatament en un problema DCONTS al qual les ciutats són representades per contenidors, només hi ha un robot, amb velocitat unitat, situat a la mateixa posició que el contenidor que representa la ciutat inicial i on tots els contenidors han de ser lliurats a la mateixa posició on son i tenen el mateix deadline  $C$ .

Per tant no només podem reduir DTSP\* a DCONTS, sinó que hem demostrat que, en realitat DTSP\* és un cas particular de DCONTS. DCONTS és, llavors, *NP*-complert, i el seu problema d'optimització associat, CONTS, és *NP*-hard.

## Appendix E

# SPARTAN in the Second ART Testbed Competition

### E.1 SPARTAN in the Second ART Testbed Competition

SPARTAN participated in the Second International ART competition, held in Hawaii, Honolulu, from May, 14 to May, 18 of 2007 in conjunction with the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007). It qualified fifth in the preliminary round and then won a position in the final round to finish in fourth place<sup>1</sup>.

Table E.1 presents the list of the contestant teams in the championship. There is a total of eighteen teams coming from universities and research centers all over the world. Tables E.2 and E.3 show the results of the preliminary and the final round of the championship, respectively.

---

<sup>1</sup>SPARTAN did not compete in the First International Competition. SPARTAN also managed to attain, without major modifications and despite major changes in the testbed (mainly relative to the number of agents in each simulation), the sixth place in the third international ART competition, held in Estoril, Portugal, in conjunction with the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008).



**Table E.1:** *List of competitors in the Second International ART Testbed Championship.*

<b>Agent Name</b>	<b>Team Representative</b>	<b>Team Affiliation</b>
ZcCariocaLes	Andrew Diniz	Pontificia Universidade Católica do Rio de Janeiro
Rex	Kunyuan Goh	Department of Computer Science, University of Warwick
Novel	Alberto Caballero Martínez	Dpto de Ingeniería de la Información y las Comunicaciones, Univ de Murcia
Blizzard	Ozgur Kafali	Department of Computer Engineering, Bogazici University
Xerxes	Jamie Lawton	Information Directorate, U.S. Air Force Research Laboratory
Candela	Edwin Boaz	Soenaryo Nanyang Technological University
Uno	Víctor Muñoz Solà	University of Girona
Marmota	Javier Murillo Espinar	University of Girona
SPARTAN	Esteve del Acebo	Agents Research Lab, University of Girona
IAM2	W. T. Luke Teacy	Electronics and Computer Science, University of Southampton
Alatrisc	Mario Gomez	Carlos III Univ. of Madrid and Univ. of Aberdeen
LesMes	Francisco París Soriano	Carlos III, University of Madrid (GIAAA)
IMM	Javier Carbo	Univ. Carlos III de Madrid
Jam	Anil Gursel	Department of Math and Computer Science, The University of Tulsa
AgenteVicente	María Teresa Vicente	Universidad Carlos III de Madrid
Rencil	Jianshu Weng	Nanyang Technological University
TDP	Doran Chakraborty	University Of Tulsa
aGente	Mikalai Sabel	University of Trento

**Table E.2:** *Results of the preliminary round of the Second International ART Testbed Championship.*

Rank	Agent Name	Avg. Bank Balance
1	IAM2	539377
2	Jam	353700
3	Blizzard	335933
4	ZeCariocaLes	319564
5	Spartan	311777
6	ArtGente	298897
7	Uno	293324
8	Rencil	269905
9	Marmota	264356
10	Novel	229501
11	Alatriste	225276
12	Rex	211467
13	IMM	200440
14	LesMes	183655
15	AgenteVicente	181932
16	Xerxes	148610

**Table E.3:** *Results of the final round of the Second International ART Testbed Championship.*

Rank	Agent Name	Avg. Bank Balance
1	IAM2	1107270
2	Jam	743077
3	Blizzard	723971
4	Spartan	674723
5	ZeCariocaLes	578524



# Bibliography

- [1] Alfaraz Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *Proceedings of the Hawaii's International Conference on Systems Sciences, Maui, Hawaii.*, pages 4–7, 2000.
- [2] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Mathematics*, 160:781–793, 2004.
- [3] ART. The Agent Reputation and Trust testbed. <http://www.art-testbed.net/>.
- [4] R. Beckers, O.E. Holland, and Dencubourg J.L. *From Local Actions to Global Tasks: Stigmergy in Collective Robotics*, pages 181–189. Artificial Life IV. MIT Press, Cambridge, MA, 1994.
- [5] E. Berlekamp and D. Wolfe. *Mathematical Go: Chilling Gets the Last Point*. A K Peters Ltd, 289 Linden Street, Wellesley, MA 02181, USA, first edition, 1994.
- [6] J.M. Bishop. Stochastic searching networks. In *Proc. 1st IEE Conf. on Artificial Neural Networks*, pages 329–331, London, 1989.
- [7] E. Bonabeau, M. Dorigo, and G. Thraulaz. *Swarm Intelligence. From Natura to Artificial Systems*. Oxford University Press, 1st edition, 1999.
- [8] E. Bonabeau and G. Thraulaz. Swarm smarts. *Scientific American*, March 2000:72–79, 2000.
- [9] R. Byrne and A. Whiten. *Machiavellian Intelligence : Social Expertise and the Evolution of Intellect in Monkeys, Apes, and Humans (Oxford Science Publications)*. Oxford University Press, USA, September 1989.
- [10] J. Carbo, J. Molina, and J. Davila. Comparing predictions of sporas vs. a fuzzy reputation agent system. In *Proceedings of the 3rd International Conference on Fuzzy Sets and Fuzzy Systems, Interlaken, Sweden*, pages 147–153, 2002.

- [11] J. Carter, E. Bitting, and A. Ghorbani. Reputation formalization for an information-sharing multi-agent system. *Computational Intelligence*, 18:515–534, 2002.
- [12] C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *Proceedings of the International Conference on Multi-Agent Systems (ICMAS'98), Paris, France*, pages 72–79, 1998.
- [13] O. Cordon and F. Herrera. A general study on genetic fuzzy systems. In *Genetic Algorithms in Engineering and Computer Science*. John Wiley & Sons, 1995.
- [14] K. Dautenhahn. Embodiment and interaction in socially intelligent life-like agents. *Lecture Notes in Computer Science*, 1562:102–142, 1999.
- [15] K. Dautenhahn. Socially intelligent agents and the primate social brain –towards a science of social minds. In *Socially Intelligent Agents: The Human in the Loop, AAI Fall Symposium*, North Falmouth, MA, 2000. AAI Press.
- [16] K. Dautenhahn. Socially intelligent agents. the human in the loop. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31:345–348, 2001.
- [17] Josep Lluís de la Rosa, Albert Trias, Antoni Martorano, Eloi Colomeda, David Huerva, and Esteve del Acebo. Shout and act. In *Proceedings of the 2009 conference on Artificial Intelligence Research and Development: Proceedings of the 12th International Conference of the Catalan Association for Artificial Intelligence*, pages 91–100, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.
- [18] E. del Acebo. Agents, sistemes multiagent i solució distribuïda de problemes. *Research Report. Institut d'Informàtica i Aplicacions. Universitat de Girona*, 2005.
- [19] E. del Acebo and J.L. de la Rosa. A fuzzy system based approach to social modeling in multi-agent systems. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15-19, Bologna, Italy*, pages 463–464, 2002.
- [20] E. del Acebo and J.Ll. de la Rosa. Bar systems. a class of optimization algorithms for reactive multi-agent systems in real time environments. In *eNTERTAIn'06 Workshop: New Trends in Real-Time Artificial Intelligence. ECAI 2006*, 2006.

- [21] E. del Acebo and J.Ll. de la Rosa. Introducing bar systems: A class of swarm intelligence optimization algorithms. In *Proceedings of the Swarm Intelligence Algorithms and Applications Symposium. SIAAS-08. Aberdeen, Scotland, 2008*.
- [22] Esteve del Acebo, Nicolás Hormazábal, and Josep Lluís de la Rosa. Appraisal variance estimation in the art testbed using fuzzy corrective contextual filters. In *Artificial Intelligence Research and Development, Proceedings of the 10th International Conference of the ACIA, CCIA 2007, Sant Julià de Lòria, Andorra, pages 104–111, 2007*.
- [23] Esteve del Acebo, Albert Oller, Josep Lluís de la Rosa, and Antoni Ligeza. Statistic criteria for fuzzy systems quality evaluation. In *Tasks and Methods in Applied Artificial Intelligence, proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE-98, Castellón, Spain, June 1-4, Volume II, pages 877–887, 1998*.
- [24] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, 2004.
- [25] B.R. Duffy. Towards social intelligence in autonomous robotics: A review. In *Robotics, Distance Learning and Intelligent Communication Systems 2001 (RODLICS 2001)*, 2001.
- [26] B. Edmonds. The constructibility of artificial intelligence (as defined by the turing test). *Journal of Logic, Language and Information*, 9(4):419–424, 2000.
- [27] B. Edmonds and K. Dautenhahn. Social embeddedness: Origins, occurrence and opportunities. In *SAB'02: From Animals to Animals 7. The Seventh International Conference on the simulation of adaptive behavior*, Cambridge, MA, 2002. MIT Press.
- [28] A.P. Engelbrecht. *Fundamentals in Computational Swarm Intelligence*. John Wiley and Sons, 2006.
- [29] B. Esfandiari and S. Chandrasekharan. On how agents make friends: Mechanisms for trust acquisition. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada, pages 27–34, 2001*.
- [30] A.S. Fraenkel and D. Lichtenstein. Computing a perfect strategy for  $n \times n$  chess requires time exponential in  $n$ . In *Proceedings of the 8th Colloquium on Automata, Languages and Programming*, pages 278–293, London, UK, 1981. Springer-Verlag.

- [31] S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *ECAI '96: Proceedings of the Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages*, pages 21–35, London, UK, 1997. Springer-Verlag.
- [32] K. Fullam, T. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. Rosenschein, L. Vercouter, and M. Voss. A Specification of the Agent Reputation and Trust (ART) Testbed: Experimentation and Competition for Trust in Agent Societies. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, pages 512–518. ACM Press, July 2005. Available at <http://www.lips.utexas.edu/art-testbed/publications.htm>.
- [33] K. Fullam, T. Klos, G. Muller, J. Sabater, Z. Topol, K. S. Barber, J. Rosenschein, and L. Vercouter. A Demonstration of the Agent Reputation and Trust (ART) Testbed: Experimentation and Competition for Trust in Agent Societies. In *The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*, pages 50–62. ACM Press, July 2005. Available at <http://www.lips.utexas.edu/art-testbed/publications.htm>.
- [34] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, January 1979.
- [35] N. Gilbert and K.G. Troitzsch. *Simulation for the Social Scientist*. Open University Press, 2005.
- [36] S. Gorard. Revisiting a 90-year-old debate: the advantages of the mean deviation. In *British Educational Research Association Annual Conference*, September 2004.
- [37] L.M. Hogg and N.R. Jennings. Socially rational agents. In *Proc. AAAI Fall symposium on Socially Intelligent Agents*, pages 61–63, Boston, Mass., 1997. Springer-Verlag.
- [38] J.S.R. Jang, C.T. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing*. MATLAB Curriculum Series. Prentice-Hall, first edition, 1997.
- [39] J.F. Kihlstrom and N. Cantor. *The Handbook of Intelligence*, chapter Social Intelligence, pages 359–379. Cambridge University Press, Cambridge, U.K., 2000.
- [40] D. Kinkielde. On the consistency of fuzzy knowledge bases. In *Proceedings of the European Symposium on the Validation and Verification of KnowledgeBased Systems*, 1993.

- [41] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice-Hall International, 1992.
- [42] H. Kummer, L. Daston, G. Gigerenzer, and J. Silk. *The social intelligence hypothesis*, pages 67–81. *Human by Nature: Between Biology and Social Sciences*, vol. 47. Erlbaum Associates, Hillsdale, NJ, 1991.
- [43] M.A. Lewis and G.A. Bekey. The behavioral self-organization of nanorobots using local rules. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1992.
- [44] S.P. Marsh. *Formalizing trust as a Computational Concept*. University of Stirling, Department of Computer Science, ph.d dissertation edition, 1994.
- [45] Merriam-Webster. Merriam-Webster online dictionary, 2007. <http://www.m-w.com/>.
- [46] P. Panzarasa and N.R. Jennings. Collective cognition and emergence in multi-agent systems. In R. Sun, editor, *Cognition and Multi-Agent Interaction: From cognitive modeling to social simulation*, pages 401–408. Cambridge University Press, 2006.
- [47] K.E. Parsopoulos and M.N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Neural Computing*, 1 (2-3):235–306, 2002.
- [48] J.A. Plucker. Human intelligence: Historical influences, current controversies, teaching resources. Retrieved 3-3-2007, from <http://www.indiana.edu/intell>, 2003.
- [49] H. Prendinger and M. Ishizuka. Let’s talk! socially intelligent agents for language conversation training. *IEEE Transactions on System, Man and Cybernetics - Part A. Systems and Humans*, 31(5), September 2001.
- [50] M. Resnick. *Turtles, Termites and Traffic Jams, Explorations in Massively Parallel Microworlds*. MIT Press, 1997.
- [51] J. Sabater. *Trust and Reputation for Agent Societies*. Monografies de l’Institut d’Investigaci en Intelligencia Artificial (20). Consell Superior d’Investigacions Cientifiques, Institut d’Investigaci en Intelligencia Artificial, Bellaterra, Catalonia, Spain, first edition, 2003.
- [52] A. Sala. Thre inference error minimization approach to fuzzy inference and knowledge analysis. In *Symposium on Qualitative System Modelling, Qualitative Fault Diagnosis and Fuzzy Logic Control*, 1996.



- [53] Michael Schillo, Petra Funk, and Michael Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence*, 14:825–848, 2004.
- [54] S. Sen and N. Saija. Robustness of reputation-based trust: Boolean case. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-02), Bologna, Italy*, pages 288–293, 2002.
- [55] E.L. Thorndike. Intelligence and its use. *Harper's Magazine*, 140:227–235, 1920.
- [56] E. Turunen. *Advances in Soft Computing*, chapter Mathematics Behind Fuzzy Logic. Physica-Verlag Heidelberg, 1999.
- [57] Li-Xin Wang. *Adaptive Fuzzy Systems and Control. Design and Stability Analysis*. Prentice-Hall, 1994.
- [58] B. Yu and M.P. Singh. Towards a probabilistic model of distributed reputation management. In *Proceedings of the Fourth Workshop on Deception, Fraud and Trust in Agent Societies, Montreal, Canada*, pages 125–137, 2001.
- [59] B. Yu and M.P. Singh. An evidential model of distributed reputation management. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems (AAMAS-02), Bologna, Italy*, pages 294–301, 2002.
- [60] G. Zacharia. *Collaborative reputation mechanisms for online communities*. Massachusetts Institute of Technology, ph.d dissertation edition, 1999.
- [61] L.A. Zadeh. *Readings in Fuzzy Sets for Intelligent Systems*, chapter Fuzzy Sets. Morgan Kaufmann Publishers, 1993.