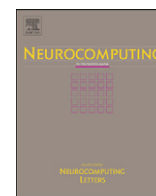




ELSEVIER

Contents lists available at [SciVerse ScienceDirect](http://SciVerse.ScienceDirect.com)

## Neurocomputing

journal homepage: [www.elsevier.com/locate/neucom](http://www.elsevier.com/locate/neucom)

## Using genetic algorithms to create solutions for conflict resolution

Davide Carneiro<sup>1</sup>, Paulo Novais\*, José Neves<sup>2</sup>

Universidade do Minho, Departamento de Informatica, Campus of Gualtar, 4710-057 Braga, Portugal

## ARTICLE INFO

Available online 9 October 2012

## Keywords:

Genetic algorithms  
Negotiation  
Online dispute resolution  
Intelligent environments

## ABSTRACT

The process of devising solutions for conflict resolution generally configures a challenging task. There exist different approaches to address the problem, namely the use of case-based models or even relying on the parties themselves to perform the task. From a computational point of view, these problems generally represent a NP-complete problem. In order to surpass this shortcoming, in this paper it is presented a biologically inspired method to deal with the problem in which genetic algorithms are used to create possible solutions for a given dispute. The approach presented is able to generate a broad number of diverse solutions that cover virtually the whole search space for a given problem. This approach provides better results than a case-based approach since: (1) it is independent of the legal domain and (2) it does not depend on the number and quality of cases present in a database. The results of this work are being applied in a negotiation tool that is part of the UMCourt conflict resolution platform.

© 2012 Elsevier B.V. All rights reserved.

## 1. Introduction

The issue of conflict resolution is a pertinent one as the number of disputes waiting to be solved in courts increases. On the one hand, there is an add to the transaction volume of global B2C e-Commerce, which leads to an increase in disputes. On the other hand, the current economic crisis led to a massive raise in the dismissals, which results in a boost in disputes related to labor law. There are also other factors that are currently contributing to the augment in the number of disagreements, which include bankruptcy, patents, information theft, real estate, consumer credit or privacy. Moreover, a major change in the paradigm is taking place in the sense that paper-based contracts, signed in the physical presence of the parties, are no longer the rule. Now, we talk of electronic contracting performed in part or wholly by means of electronic agents and over a telecommunication system. Obviously, the courts, shaped after the industrial era and still paper-based, are not ready for the new characteristics of these disputes.

In an attempt to deal with this problem, different approaches to litigation in courts were adopted, namely the so-called Alternative Dispute Resolution (ADR) methods that involve, among others, negotiation, mediation or arbitration [1,5,7]. With the advent of the Information Society, these techniques are being

considered of use in virtual environments, leading to what is known as Online Dispute Resolution (ODR) [2]. Indeed, ODR implements traditional debate methods over a communication mean, i.e., instead of negotiating in person, the disputant parties do it over a phone line or any other communication mean. However, the latest research trends show that the role of technology in dispute resolution can be further enhanced, namely by using techniques and methodologies for problem solving taken from the Artificial Intelligence arena [2]. In such ODR systems, technology will be used not only to put parties into contact but also to suggest solutions, strategies or to compile useful information.

The most common way of implementing ODR tools is by means of Expert Systems. As a result, two main trends can be identified, namely Rule-based and Case-based ODR tools [9]. The use of Rule-Based Systems (RBS) aims to encode the problem solving expertise of human connoisseurs, which may be seen as a set of procedures guided by rules [10]. The use of Case-based Reasoning (CBR), on the other hand, aims at capturing and using knowledge from past experiences, allowing known solutions to be applied to solve similar problems, either in the present or the future [11].

RBS models were first used to develop Expert Systems for the legal domain. Generally, a rule describes the conclusions that the expert draws from a set of facts. The whole set of rules constitutes the knowledge base about the domain. Then, when the system is provided with facts, it searches the knowledge base for the relevant rules and applies them in order to reach a conclusion. This kind of systems builds on the idea that chains of rules can define the reasoning processes of human experts [12]. Known projects following this trend include TAXMAN, presented in 1977 by McCarty [13], LDS, by Waterman and Peterson [14], and Popp and Schlink's JUDITH system [15].

\* Corresponding author. Tel.: +351 253 604 437; fax: +351 253 604 471.

E-mail addresses: [dcarneiro@di.uminho.pt](mailto:dcarneiro@di.uminho.pt) (D. Carneiro),[pjon@di.uminho.pt](mailto:pjon@di.uminho.pt) (P. Novais), [jneves@di.uminho.pt](mailto:jneves@di.uminho.pt) (J. Neves).URLs: <http://islab.di.uminho.pt/davide> (D. Carneiro), <http://www.di.uminho.pt/~pjn/> (P. Novais), <http://www.di.uminho.pt/~jmn/> (J. Neves).[www.di.uminho.pt/~pjn/](http://www.di.uminho.pt/~pjn/) (P. Novais), <http://www.di.uminho.pt/~jmn/> (J. Neves).<sup>1</sup> Tel.: +351 253 604 435.<sup>2</sup> Tel.: +351 53 604 466.

In the last years, however, the trend has been changing towards the use of a CBR approach. The fundament of CBR is to store and index key information of known cases. These indexes may then be used to search for past cases that may be of relevance to solve a current problem. The key idea is that similar solutions applied to similar problems in a similar context will lead to similar and predictable outcomes [16]. CBR may be used to improve problem-solving tasks in quite a few ways. Namely, it can provide interpretation shortcuts, warn for potential errors or suggest lines of thought [17]. There is extensive work in the field of CBR applied to the legal domain. Some of the most well-known projects include the MEDIATOR [17], the PERSUADER [18], or James Popple's SHYSTER [19].

However, both rule-based and case-based approaches to problem solving in the legal arena face criticism, given their prospective disadvantages. First of all, most of the systems developed so far are restricted to specific domains of law. This makes it hard to reproduce the results of a given project in other domains.

Considering RBSs, the main disadvantages of this approach are linked with the human experts that formulate the rules. In fact, a big challenge is effectively to determine the *Quality-of-Information (QoI)* coded by the rules. In that sense, there must be a big concern in involving experts from both the Legal and/or the Computer Science arenas in the task of defining the rules. Moreover, a RBS is not an optimal solution for all problems. Consequently, considerable knowledge is needed not to misapply these systems. Finally, it is a fact that the ease of rule creation and edition can be an advantage. However, this can be also seen as a potential disadvantage as the system can easily be sabotaged by a non-knowledgeable user. Typical reasons for the failure of RBSs include the negligence to employ simple tools for system audit that can detect conflicts in rules.

Considering CBR, its main disadvantage stems from its potential complexity. In fact, in a multifaceted knowledge-based domain, the simple definition of the content of a case may be an overwhelming task, involving many different experts from different fields of knowledge. Moreover, this also implies the use of significant amount of resources to store and deal with such cases. This means that case-based approaches are generally more intricate and resource demanding. Additionally, most of the analyzed systems are static rather than adaptive. This means that once a strategy is defined, generally at the outset of the process, it will be followed disregarding probable changes in the environment that sets the context of interaction. We can thus succinctly enumerate the main drawbacks of both RBS and CBR approaches as follows:

- Laws change constantly thus implying updates to rules in RBS. This frequently results in inconsistencies and/or redundancy. Moreover, this might be a quite complex task (depending on the complexity of the legal domain) that must be performed manually.
- The quality of a RBS ODR tool is directly dependent on the quality of the work of the humans translating the legal norms into rules. The quality of information of the rules may be hard to determine.
- RBS are static and will not shape changes in the legal domain, unless these are coded manually by a human expert.
- The quality of a CBR ODR tool is directly dependent on the quality and amount of past cases known.
- The fact that legal norms change frequently also has a negative impact in CBR approaches, rendering past cases potentially useless under the light of the new norms.
- Both CBR and RBS approaches are domain dependent. This implies that rules are defined independently for each legal domain and that cases from a specific domain can hardly be reused to another.

Another disadvantage of current ODR tools is the loss of context information regarding the context of interaction. This information can be described as the state of the parties and includes parameters such

as the level of stress, the level of escalation, the emotional state or the stress, just to name a few. All this information is taken into account by a judge or a jury in litigation in court or by a neutral controlling an alternative process, but is lost when using an ODR tool. Although it can be argued that legal disputes are more about reason and evidence and less about emotions, these ultimately influence all our decisions and actions and should be regarded as very important. As an example, a party that is very stressed or feeling extremely anger at a given time in the dispute resolution should not take important or binding decisions as he might regret them later. Instead, he should be advised to make a pause and think about it later on. The development of ODR systems that are indeed able to understand the emotional state of the different parties is thus interesting.

Recently, soft computing techniques have been used to address this kind of complex problems. Take as example the works described in [3,4], in which soft computing techniques are used to detect the lifetime of building thermal insulation failures and to identify typical meteorological days, respectively. More specifically, evolutionary computation has been successfully used to deal with problems that involve a significant amount of complex variables in which traditional approaches would not be suited (e.g. [25,26]). Following this approach, in this work, we use genetic algorithms to generate solutions for the resolution of a conflict. The main objective is to assist parties and mediators in generating and modifying consensual solutions in a conflict resolution process, generally a negotiated one. The presented approach has, in our opinion, three main advantages:

- It is a more complete solution when compared with case-based approaches, as it may cover virtually the entire problem search space.
- The number of generated solutions depends only on the initialization settings.
- It is domain-independent, contrary to case-based approaches in which cases can generally only be reused in the same legal domain.

To evaluate this approach we are reusing the UMCourt conflict resolution platform [20,22], by replacing the solution generation module, that was case-based, by the one described in this work [21]. This work is being carried out under the TIARAC project—Telematics and Artificial Intelligence in Alternative Conflict Resolution Project (PTDC/JUR/71354/2006), a research project supported by the Portuguese Science and Technology Foundation. The remaining of the paper is structured as follows. Section 2 presents the core of the paper, in which the use of genetic algorithms to create solutions for conflict resolution is depicted, including detailed descriptions of the genetic operators. In Section 3 we describe the main domain of application of this work, i.e., a negotiation environment powered by the genetic algorithm described in Section 2. Finally, in Section 4 we describe some of the results and Section 5 synthesizes the main conclusions.

## 2. Using genetic algorithms to create solutions for conflict resolution

Engendering solutions for the resolution of a conflict is generally a challenging task, although different methods may be used. The employment of case-based models may be a good solution as conflict resolution experts themselves rely on past experiences to make their judgments. However, a poor or unfit case-base may disrupt the system. On the other hand, relying on the parties themselves to perform this task is, in general, not a good approach since they may be unwilling or unable to do so. In order to deal with this drawback, we propose the use of genetic algorithms to create solutions for conflict resolution.

$$P = \left[ \begin{array}{c|c|c|c} Ch_1 = \begin{bmatrix} V_{1,1} & \cdots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{m,1} & \cdots & V_{m,n} \end{bmatrix} & Ch_2 = \begin{bmatrix} V_{1,1} & \cdots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{m,1} & \cdots & V_{m,n} \end{bmatrix} & \dots & Ch_s = \begin{bmatrix} V_{1,1} & \cdots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{m,1} & \cdots & V_{m,n} \end{bmatrix} \end{array} \right]$$

Fig. 1. Under this model a population of size  $s$  is represented as a set of chromosomes with a cardinality of  $s$ .

Under this approach, each chromosome will represent a possible solution for a conflict, i.e., a possible distribution of the items in dispute. The population evolves from generation to generation by means of genetic operators that change the items distribution and thus have an effect on the fitness of the solutions. This fitness is computed from the point of view of each party, i.e., a solution that is good for one party may not be so good for another party, given that they have conflicting objectives. With the successive generations, some chromosomes tend to be more fit to one party, i.e., there are lines of evolution that evolve naturally towards the maximization of the fitness of a given party. In this work, a species is thus defined as the group of chromosomes that stand for good solutions for a given party. Thus, there will be a species for each party. Moreover, a chromosome may belong to more than one species if it denotes a solution that is good for more than one party. Evidently, chromosomes that belong to more than one species are more attractive, since they correspond to solutions that will be more easily accepted by the parties.

## 2.1. Chromosome

A population  $P$  of size  $s$  is defined by a set of chromosomes  $Ch$  (Fig. 1). Each chromosome  $Ch_i$ ,  $i \in 1, 2, \dots, s$  represents a possible solution for the problem, i.e., a chromosome represents a distribution of the items in dispute or, in other words, who gets how much of what. For a dispute involving  $n$  parties and  $m$  issues, a chromosome  $Ch$  can be represented as an  $m$ -by- $n$  matrix (Eq. (1)).

$$Ch = \begin{bmatrix} V_{1,1} & \cdots & V_{1,n} \\ \vdots & \ddots & \vdots \\ V_{m,1} & \cdots & V_{m,n} \end{bmatrix} \quad (1)$$

Under this representation, the value  $V_{m,n}$  denotes the amount of issue  $m$  that the party  $n$  receives in terms of the solution  $Ch$ . The actual content of the chromosome depends on the domain of the dispute. Moreover, each domain also has specific rules that ensure the correctness of the solution. Let us take as an example the general model of distributive negotiation, in which a group of resources must be divided (e.g. divorces, winding up of a company). Under this model each entry stands for a value between 0 and 1 Eq. (2), and the sum of the values of each line  $m$  be 1 Eq. (3). The total amount of resources received by party  $n$ ,  $R_n$ , is defined as the sum of the values of column  $n$  Eq. (4).

$$V_{m,n} \in A, \quad A = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\} \quad (2)$$

$$\sum_{i=1}^n V_{m,i} = 1, \quad \forall m \in \{1, 2, \dots, m\} \quad (3)$$

$$R_n = \sum_{i=1}^m V_{m,n} \quad (4)$$

This model also considers the existence of indivisible goods, i.e., goods that cannot be divided. This is common in scenarios such as divorces, in which there are assets (e.g. car, house) that parties are not willing to sell to split the money, and must be assigned to one party only. Thus, for every indivisible issue  $m$ , Eq. (5) must hold true.

$$V_{m,i} = 1 \Rightarrow V_{m,x} = 0, \quad \forall x \in \{1, 2, \dots, n\}, \quad x \neq i \quad (5)$$

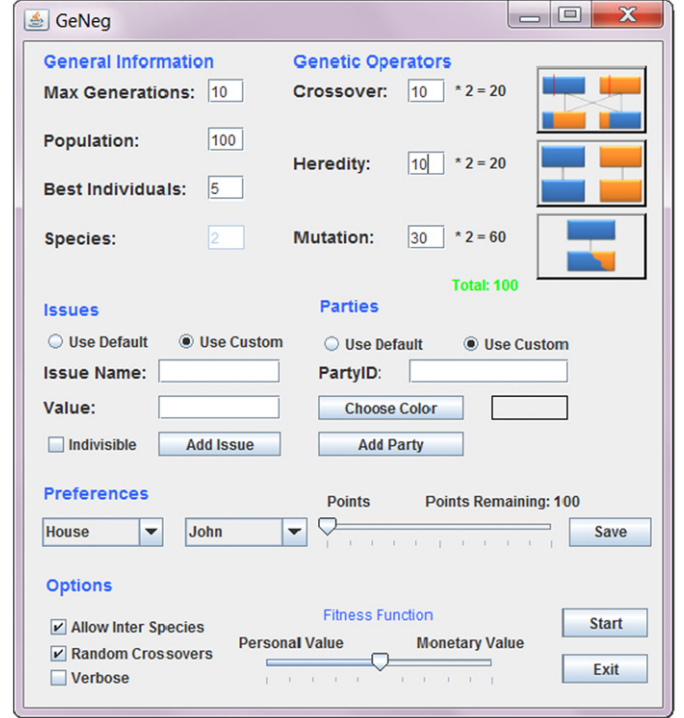


Fig. 2. Prototype of the interface used to configure the genetic algorithm, including information about the parties, the issues and the weight of each genetic operator.

## 2.2. Initialization

Initially, some information must be provided to the system in order for it to be initialized (Fig. 2). Concerning the conflict resolution process in itself, the name, value and type of each issue under negotiation must be provided, as well as a name and a color for each party. Moreover, each party must state its preferences concerning each item in dispute by distributing 100 points among the several items. This will allow the system to determine how much each party values each item, from a personal subjective perspective, and not a monetary one.

In terms of the genetic algorithm, some information is also needed for the initialization process. A predefined number of runs sets the termination condition. The size of the population stands for the number of chromosomes at each running cycle. Checking the *verbose* option will make the algorithm detail all its steps. The algorithm can also be configured in terms of how many individuals are considered from each species, to which the genetic operators are applied, being that a higher number of individuals will result in a higher diversity. Concerning the genetic operators, it is also possible to define the weight of each one in the making of new populations, i.e., it is possible to decide on how many individuals of the new population will be generated by each operator. Finally, it is also possible to configure the fitness function in terms of the weights of the personal and monetary values in the computation of the overall fitness of each solution. Once this information is provided, the algorithm can be initialized. In this process, a population of the specified size is generated,

where each chromosome is set in terms of a random distribution of the items, i.e., a solution generated randomly.

### 2.3. Selection

At each running cycle there is a part of the population that is selected from each species to give birth to a new one. Given that only the best individuals are selected to breed the new population, a fitness-based process must be defined to set which of the fitter solutions of each species may be pointed out. As each solution has a different fitness for each party, the fitness of each solution for each party must be computed. This means that for a conflict resolution involving  $n$  parties and for a population of size  $s$ ,  $n*s$  values of fitness will be computed at each running cycle.

The fitness function looks at each solution and assigns it a value that depends on the amount of items that each party receives, i.e., the economic value of the items and the value that each party assigned to each item, i.e., the personal value. In this work, two fitness functions were devised Eqs. (6) and (7), where

- $tmv$  denotes the case economic value, i.e., the total amount of money that the issues in dispute are worth with, being defined as  $tmv = \sum_{i=1}^I mv_i$ ;
- $I$  defines the number of issues.
- $mv_i$  stands for the monetary value of issue  $i$ .
- $fit_{j,p}$  represents the fitness of chromosome  $j$  for party  $p$ .
- $W_m$  denotes the weight of the monetary component while
- $W_p$  stands for the weight of the individual component.
- $prefs_i$  denotes the preferences of a given party regarding issue  $i$ .

The use of Eq. (6) will tend to result in solutions in which each party receives approximately what he valued the most, i.e., Eq. (6) minimizes the difference between the individual preferences for each item and the personal value assigned to it.

$$fit_{j,p} = W_m * \frac{\sum_{i=1}^I Ch_{j,p} * mv_i}{tmv} + W_p * \left( 1 - \frac{\sum_{i=1}^I |Ch_{j,p} - prefs_i|}{I} \right) \quad (6)$$

On the other hand, Eq. (7) focuses on maximizing the individual gain of each party, i.e., Eq. (6) will tend to generate populations that may be described as fair, while the ones generated by Eq. (7) will be more greedy.

$$fit_{j,p} = W_m * \frac{\sum_{i=1}^I Ch_{j,p} * mv_i}{tmv} + W_p * \sum_{i=1}^I \frac{|Ch_{j,p} - prefs_i|}{I} \quad (7)$$

The system is currently using Eq. (6) as it generates solutions that, by being more balanced, are more likely to be accepted by the parties. At each cycle, the fitness of the population is computed and the best individuals from each species are selected to generate a new population.

### 2.4. Reproduction

In genetic algorithms, reproduction looks at the engendering of new populations, making the heuristic search move towards the maximization of the fitness function. In this work three different genetic operators are being used: mutation, crossover and heredity. They are applied to each population according to what was defined during the initialization process.

#### 2.4.1. Mutation

In genetics, a mutation is defined as a spontaneous and random change in a genomic sequence. In the particular case of this work, a mutation is a random change at the chromosome level, i.e., in terms of the issues distribution. The issues mutated are given in terms of a mutation threshold, here designated as  $\mu$ . To implement

the mutation, a random issue is selected as well as two random parties. The allocation of the issue is then changed for the selected parties, according to the mutation threshold. If the issue is divisible, the amount of the issue is subtracted from one party according to  $\mu$  and added to the other party. If the issue is indivisible, there is a probability that a change of ownership may occur, according to the mutation threshold. Once a new chromosome is generated its validity is checked in order to determine if all the invariants hold. If the chromosome is not valid (e.g. contains a negative value) it must be generated again. This process repeats until the chromosome is valid (which eventually happens). Let us consider the scenario in which two parties, with a given  $\mu$ , are disputing four issues. Assuming that issue 2 is divisible and is selected to be changed between party1 and party2, the result is shown in Eq. (8), where  $Ch$  and  $Ch'$  present a picture of a chromosome, respectively, before and after the operation.

$$Ch = \begin{bmatrix} V_{1,1} & V_{1,2} & V_{1,3} \\ V_{2,1} & V_{2,2} & V_{2,3} \\ V_{3,1} & V_{3,2} & V_{3,3} \\ V_{4,1} & V_{3,2} & V_{4,3} \end{bmatrix} \quad Ch' = \begin{bmatrix} V_{1,1} & V_{1,2} & V_{1,3} \\ V_{2,1} + \mu & V_{2,2} - \mu & V_{2,3} \\ V_{3,1} & V_{3,2} & V_{3,3} \\ V_{4,1} & V_{3,2} & V_{4,3} \end{bmatrix} \quad (8)$$

The mutation, as defined here, has an effect on the fitness of the solution for each party. That is, the new solution will be more favorable to party1 and less favorable to party2. Below the mutation algorithm is described.

Algorithm Mutation is

Input: List of parties, L

List of issues, I

Parent chromosome, C

Output: A new chromosome, C'

Do

$i :=$  select random issue from I

$p1 :=$  select random party from L

$p2 :=$  select random party from L such that  $p1 \neq p2$

$C' := C$

if (i is divisible)

$C'_{i,p1} := C'_{i,p1} + * C'_{i,p1}$

$C'_{i,p2} := C'_{i,p2} - * C'_{i,p2}$

else if (randomNumber >  $\mu$ )

temp :=  $C'_{i,p1}$

$C'_{i,p1} := C'_{i,p2}$

$C'_{i,p2} := temp$

While (C' is invalid solution)

Return C'

#### 2.4.2. Crossover

Crossover is a binary genetic operator by means of which two offspring are generated from two parent chromosomes. In this work, a two-point crossover technique is used. Basically, these two points define the beginning and the end of an issue for both parents, meaning that everything between the beginning and the end of the issue will be swapped to create two offspring, i.e., crossover consists in swapping two distributions of the same issue, generating two new solutions. Two lines of attack can be used in this technique that will influence the variety of the new offspring: *inter-species* and *random parents*. If the *inter-species* option is set to *true*, this means that chromosomes from different species can be crossed, generating a more diverse population. The *random parents* option will allow one to decide on which parents should be used. If it is set to *true*, random parents will be selected. On the other hand, if it is set to *false*, the best parents of each species will be selected for the crossover operation. Eq. (9) depicts an example of the use of crossover in two parent chromosomes  $Ch1$  and  $Ch2$  to generate two offspring  $Ch1'$  and  $Ch2'$ . In this

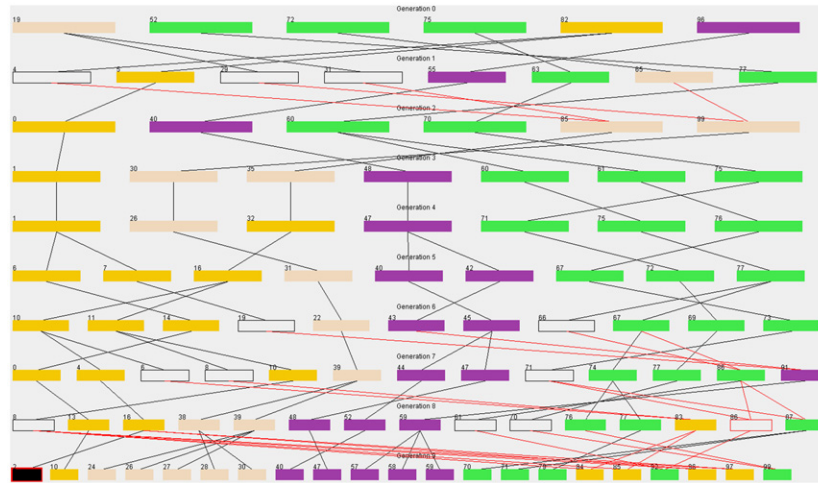


Fig. 3. Lines of evolution of the genetic course and their outcome.

example the distribution of issue 2 was swapped.

$$\begin{aligned}
 Ch1 &= \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \\ J & K & L \end{bmatrix} & Ch2 &= \begin{bmatrix} M & N & O \\ P & Q & R \\ S & T & U \\ V & W & X \end{bmatrix} & Ch1' &= \begin{bmatrix} A & B & C \\ P & Q & R \\ G & H & I \\ J & K & L \end{bmatrix} \\
 Ch2' &= \begin{bmatrix} M & N & O \\ D & E & F \\ S & T & U \\ V & W & X \end{bmatrix} & & & & (9)
 \end{aligned}$$

Given that this technique changes the distribution of each solution, it will have effect on the fitness function. Below it is given a description of the generic algorithm that implements the crossover technique being used here.

```

Algorithm Crossover is
Input: List of parties, L
         List of issues, I
Output: New chromosomes, C1', C2'
i := select random issue from I
if (interspecies)
    s1 = select random species
    s2 = select random species such that s1 != s2
    if (randomparents)
        C1 := select random ch from s1
        C2 := select random ch from s2
    else
        C1 := select best ch from s1
        C2 := select best ch from s2
else
    s1 = select random species
    if (randomparents)
        C1 := select random ch from s1
        C2 := select random ch from s1 such that C1 != C2
    else
        C1 := select best ch from s1
        C2 := select second best ch from s1
swap issues and generate C1', C2'
return C1', C2'
  
```

#### 2.4.3. Heredity

In genetics, heredity can be defined as the passing of traits from parent to offspring. Once this process is finished, the offspring acquires characteristics that may be compared to the ones of the parent. The evolution of the species is thus achieved

by accumulating variations exhibited by different individuals. In this work, heredity is the simplest genetic operator to be considered, in the sense that the new offspring have exactly the same distribution that its parents, i.e., this operator generates no diversity at all and should be used when one wants to avoid “losing” the best individuals of a population, ensuring that the best traits will be passed to the next generation.

#### 2.5. Termination

The process of selection and reproduction is repeated until the termination condition is reached, i.e., the maximum number of generations stated in the initialization phase is reached. At this point, the system may give a picture of the state of the evolutionary process, in terms of the solutions so far attained, and their lines of evolution (Fig. 3). Moreover, each solution is represented with one or more colors. A solution with a given color means that it belongs to the species (party) of that color. This will allow one to see the natural emergence of species, i.e., the lines of evolution that tend towards the maximization of the fitness value for a given party. Colorless individuals denote solutions that are not among the most fit for a particular population but generate offspring that are among the best of future ones. It is also possible to look at a chromosome's content, as well to its fitness, and the fitness mean, by clicking on it. The lines between individuals stand for the parent–offspring relationships. A unary genetic operator generated an individual that has a single line connecting to the previous generation, while an individual that has two lines was generated by crossover.

### 3. A bio-inspired negotiation environment for conflict resolution

The genetic algorithm presented in this work amounts to a way of making possible solutions for a given dispute. In reality, we are modifying the negotiation algorithm that relied on a case-based approach, so that it proposes solutions generated by the genetic algorithm referred to above. The main advantage is that this method is domain-independent and the solutions cover virtually all the search space.

It is being developed a negotiation setting that is based on the original case-based algorithm but whose solutions are proposed by the genetic one. All the processes are conducted by a human mediator that is supported by the assistance of what can be called as an electronic mediator: an autonomous software agent that

supports the decision-making tasks of a human mediator by providing useful information [8]. To better support the task of the mediator, we are developing the concept of conflict resolution environment. This is based on insights from Ambient Intelligence and aims to provide the mediator with context information about the conflict resolution environment that may be paramount to take rational decisions, in a model in which the parties are not in his physical presence [23]. At the moment, the mediator has access to information about the parties' conflict resolution style and their level of stress.

The personal conflict style [24] characterizes the way a person handles a conflict. Once each person has a different way of acting in a conflict situation, having access to this information is important to plan effective conflict resolution strategies [6]. In fact, the system is constantly monitoring the behavior of the parties, gathering information about their interactions, in order to infer their conflict style [22], i.e., the system analyzes if each party is contributing with proposals (e.g. proposing new solutions, answering to proposed solutions) or if they are not cooperating (e.g. ignoring proposals, always answering negatively without making counter-proposals). In addition, if a party is actually cooperating, the system tries to analyze the intention of the party. This is done by looking at the utility of the proposal for each party. If a party is constantly proposing solutions that focus only on the maximization of his personal gain, he is not collaborating towards the conflict resolution at all, as that party is focusing on the maximization of its personal gain. On the other hand, if the utility of the proposed solution is more balanced, that shows that the party is willing to make some trade-offs, to lose something, in order to solve the problem. The relation between the utility of the proposals and the conflict style is depicted in Fig. 4. In a few words, the utility of a proposal is analyzed according to the values of the BATNA and WATNA (Best and Worst Alternative to a Negotiated Agreement, respectively) and to the ZOPA (Zone of Potential Agreement). A detailed description of how the system computes and provides information about the personal conflict styles of the parties to the mediator is given in [22,27].

Taking into consideration this type of information, the system classifies the personal conflict style of each party in real-time as competing, accommodating, avoiding, collaborating or compromising. It is then possible to see the evolution of the conflict style throughout the negotiation process and a mediator will be able to identify trends or typical behaviors (e.g. a party that starts in an avoiding style but, as confidence builds, moves to a compromising

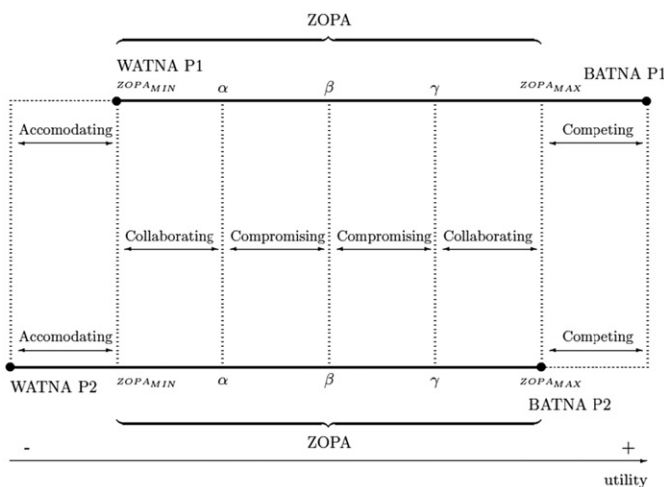


Fig. 4. The space that defines the personal conflict styles in function of the utility of the proposals and the values of the BATNA, BATNA and ZOPA.

one) (Fig. 5). From the point of view of a mediator, this information is very important as it allows him more accurately guide the conflict resolution process to a successful end.

As mentioned above, further information is considered in this model, namely the level of stress of the parties. In order to determine this parameter, users are provided with mobile devices equipped with touch screens and accelerometers that are used to interact with the system (Fig. 6). From these devices we can extract information about the accuracy of the touch, the intensity of the touch, the touch pattern and the amount of movement (given from the variations in the acceleration). The amount of movement is also given from external video cameras placed in the environment. All these issues are related to stress (e.g. smaller touch accuracy and higher touch intensity are related to higher levels of stress, more and more sudden movements are related to higher levels of stress). Considering all this information, the system computes an estimation of the level of stress of each user, as described in [28].

The mediator thus has access to important information from the context of interaction that depicts the state of the parties. This will allow him to implement a dynamic conflict resolution model, in which he will be able to adapt the strategy devised at the outset of the process in real time, to respond to significant changes in the state of the parties. To support this, the system is able to perform two main operations on the populations produced, namely jumps and walks. In a jump, the system selects a solution from a different species (Fig. 7). This may happen in a scenario in which the mediator realizes that a given party is being penalized or when he is getting close to abandoning the process. In that sense, the system will jump to the species that have more constructive solutions to that party. Walks, on the other hand, are used to move within the same species (Fig. 8). This usually takes place when the state of one or more parties changes slightly. When the system moves down the tree it is searching for solutions that are better for the current party (as solutions improve as the generations increase) and vice versa.

Based on what was stated above we are now defining a dynamic negotiation model whose main objective is to be able to interpret context information and adapt strategies in real time in a conflict resolution process, i.e., to be more context-aware and adaptive. In fact, conflict resolution methods that are run by

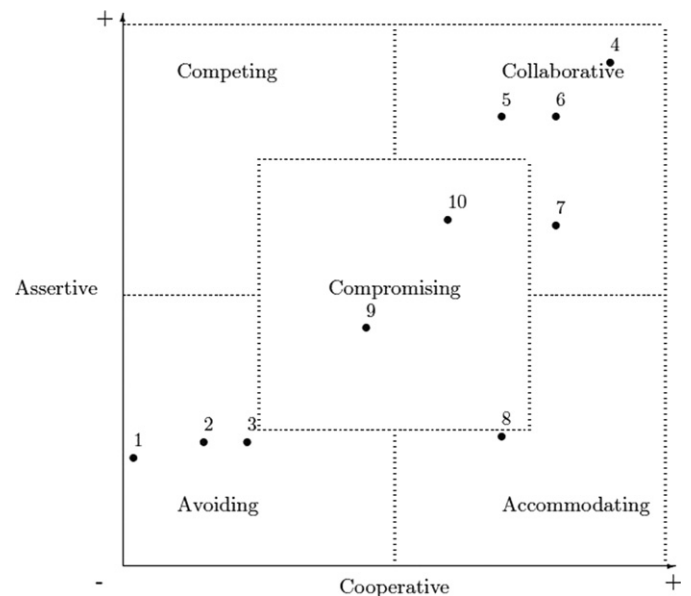


Fig. 5. An evolution of the style of conflict: the system constantly monitors the interactions of each party in order to infer, in real-time, their individual conflict modes.

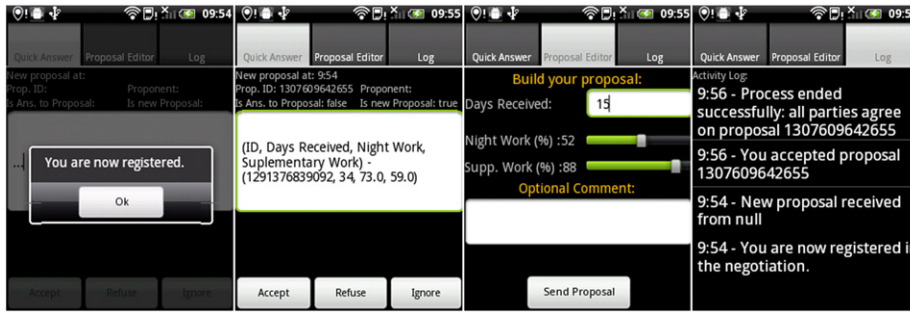


Fig. 6. Four android interfaces for mobile devices with touch screens: the former shows a message confirming the registration of the user in the negotiation session; the second shows a proposal in the quick answer screen; the third shows the proposal editor screen; and the last shows the message log.

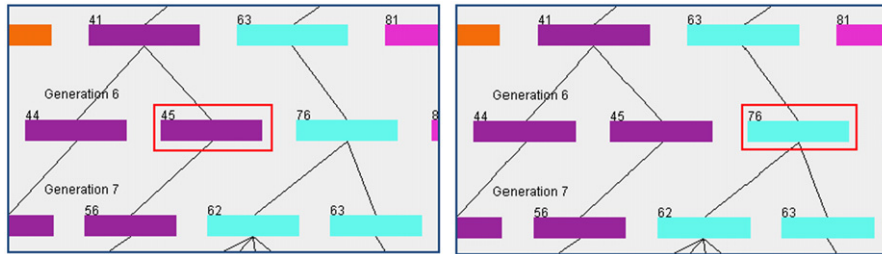


Fig. 7. A jump: when the state of another party is worsening, the mediator may select a solution from the species of that party in the next round.



Fig. 8. A walk: when the state of the party improves or worsens, the mediator is able to move up or down the tree in order to select solutions that are more suited to the latest parties state.

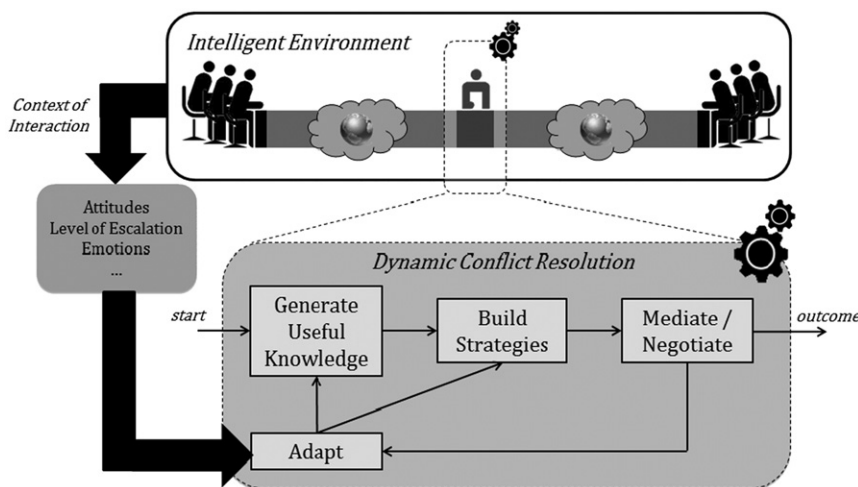


Fig. 9. A dynamic negotiation model: this picture depicts a high level view of a dynamic negotiation setting that is able to adapt strategies and propose different solutions according to changes in the process of interaction.

human experts in the physical presence of the parties are generally dynamic as experts have the ability to understand changes that may occur in the interaction framework (e.g. a party is getting

stressed, a party does not like the current state of affairs), and change the strategy before it is too late (e.g. by making a pause in the process). However, the problem is that under ODR settings, such

context information is not available for the mediators as the parties are, generally, “hidden” behind a web interface.

This dynamic negotiation or mediation process starts with a phase in which useful knowledge is generated that will be used by the parties and the mediator to build strategies, as depicted in [20]. Afterwards the actual conflict resolution process starts. The first solution proposed by the system is the solution with the highest mean value of fitness as we believe that this is the solution that will be more consensual. From this point on, parties can answer to the proposal with a new proposal, with a modification of the current one or simply be accepting, ignoring or refusing it. Whenever the mediator feels like proposing a new solution, it may do so with the support of the proposed approach, by using jumps and walks on the solution tree. These operations may happen every time a significant change in the context of interaction takes place. In future work we intend to include additional context information, such as the emotional state, inferred from the processing of the voice or of the facial expression of the parties. A high level view of this approach is depicted in Fig. 9.

#### 4. Results

In this section we show some of the results achieved so far and compare them with the previous case-based approach. In order to get these results, some negotiation scenarios with a group of our students were set-up. Each negotiation scenario involved four students arguing about four issues. The genetic algorithm was configured to allow inter-species and random crossovers. The maximum number of generations is 100 as well as the size of the population. The fitness function used assigned equal weight to the individual and economic values, an approach that seems to lead

to solutions that are fairer. The following four scenarios show the settings of four negotiation sessions as well as the best overall solution (the solution with the highest fitness mean) and the fitness of the best solution for each party.

Scenario 1. Under this setting, mutation has a significantly higher weight than crossover and heredity is not used. The fitness mean of the best overall solution is around 0.71. In fact, according to this outcome, parties 1, 2, and 3 receive the issue that they were waiting for. Furthermore, they still receive a part of the economic issue (Tables 1.1–1.3).

Scenario 2. In this scenario the same issues for each party were employed (Table 2.1) and the parties also have the same preferences (Table 2.2). However, the use of the crossover and mutation operators is switched. It may be seen that, compared to the previous scenario, the best overall solution is achieved at a higher computational cost and with a significantly smaller fitness mean (Table 2.3). However, the values of the fitness of the best solution for each species are significantly higher.

Scenario 3. In this scenario two parties want to win the same issue at all cost (Tables 3.1 and 3.2). However, as the issue is indivisible it can only be assigned to a single party. In such scenarios, the fitness mean of the best overall solution is unavoidably lower than in other scenarios (e.g. scenario 1) (Table 3.3).

Scenario 4. According to what we expected, the use of heredity, even at a small scale, does not necessarily leads to better solutions. In fact, when heredity is used, the best overall solution tends to be achieved at a higher computational, being the fitness mean lower (Tables 4.1–4.3).

The results achieved with this approach can be compared to the ones achieved using our previous case-based system. Concerning the temporal performance of both approaches, this one performs better

**Table 1.1**  
Description of scenario 1.

Name	Type	Value
Issue1	Indivisible	20,000
Issue2	Indivisible	50,000
Issue3	Monetary	30,000
Issue4	Indivisible	60,000

**Table 2.1**  
Description of scenario 2.

Name	Type	Value
Issue1	Indivisible	20,000
Issue2	Indivisible	50,000
Issue3	Monetary	30,000
Issue4	Indivisible	60,000

**Table 1.2**  
Distribution of the preferences.

	Party1	Party2	Party3	Party4
Issue1	0.5	0.1	0.1	0.2
Issue2	0.2	0.6	0.1	0.2
Issue3	0.2	0.1	0.1	0.3
Issue4	0.1	0.2	0.7	0.3

**Table 2.2**  
Distribution of the preferences.

	Party1	Party2	Party3	Party4
Issue1	0.5	0.1	0.1	0.2
Issue2	0.2	0.6	0.1	0.2
Issue3	0.2	0.1	0.1	0.3
Issue4	0.1	0.2	0.7	0.3

**Table 1.3**  
Final statistics of the run.

Genetic operators	Crossover	Heredity	Mutation	
	20	0	80	
Best overall solution	Generation:	5	Mean fitness: 0.7056	
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0.25 & 0.19 & 0.25 & 0.31 \\ 0 & 0 & 1 & 0 \end{pmatrix}$			
Fitness of best solution for each species	Party1 0.6809	Party2 0.6847	Party3 0.7043	Party4 0.7927



**Table 2.3**  
Final statistics of the run.

Genetic operators	Crossover	Heredity	Mutation	
	80	0	20	
Best overall solution	Generation: 18		Mean fitness:	0.5312
	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0.23 & 0.23 & 0.23 & 0.31 \\ 0 & 0 & 1 & 0 \end{pmatrix}$			
Fitness of best solution for each species	Party1 0.6761	Party2 0.7161	Party3 0.7161	Party4 0.6778

**Table 3.1**  
Description of scenario 3.

Name	Type	Value
Issue1	Indivisible	20,000
Issue2	Indivisible	50,000
Issue3	Monetary	30,000
Issue4	Indivisible	60,000

**Table 4.1**  
Description of scenario 4.

Name	Type	Value
Issue1	Indivisible	20,000
Issue2	Indivisible	50,000
Issue3	Monetary	30,000
Issue4	Indivisible	60,000

**Table 3.2**  
Distribution of the preferences.

	Party1	Party2	Party3	Party4
Issue1	1.0	1.0	0.2	0.1
Issue2	0.0	0.0	0.5	0.2
Issue3	0.0	0.0	0.2	0.3
Issue4	0.0	0.0	0.1	0.4

**Table 4.2**  
Distribution of the preferences.

	Party1	Party2	Party3	Party4
Issue1	0.5	0.1	0.1	0.2
Issue2	0.2	0.6	0.1	0.2
Issue3	0.2	0.1	0.1	0.3
Issue4	0.1	0.2	0.7	0.3

(generally under 1 s, depending on the initialization parameters). The previous approach would usually take several seconds as it could require several interactions with the database in order to better adapt the search queries to the amount and type of cases. However, the time is in fact not very important given that these processes are generally asynchronous and develop over several days. Indeed, to make this analysis, we prefer to focus on the amount of cases retrieved/generated by each approach. Looking at the case-based approach, the retrieval of cases fails on 66% of the requests, with failing meaning that some rule was broken (e.g. not enough cases were selected, too many cases were selected, there are not enough cases in the database). Although the system may try to recover by relaxing or strengthening search rules, it will take more time and it may finally fail when rules cannot be changed further. Moreover, when rules are changed, results may not be satisfactory (e.g. we might be selecting cases that are not similar enough for the problem). Given this, it can be stated that the quality and amount of the cases retrieved directly depends on the case-base, which may be poor. In that sense, the current approach performs better as the variety and amount of possible solutions are only limited by the initialization parameters and we can possibly cover the whole search space. Moreover, users will not be overwhelmed with a huge amount of information as the described algorithm will select, among all the generated solutions, the best to be proposed at each time, a task that is always validated by the human mediator. In this sense, this approach is more complete and more reasonable than the previous case-based one.

## 5. Conclusions

From these experiments, we can conclude that the best solutions are achieved when heredity is not used and when mutation has a

significantly higher weight than crossover. In fact, the use of heredity preserves the best trends of the population. However, it also tends to slow down evolution. Indeed, suitable solutions tend to be attained at a higher computational cost. It was also concluded that when parties have conflicting objectives and focus on a very reduced set of issues, no satisfactory solutions are achievable. From the experiment we also concluded that the best value for  $\mu$  is 0.02. In fact, using a smaller value will make the evolution of the population very slow, taking many runs to have adequate solutions. On the other hand, a higher value will make the evolution too random, making it hard to emerge a natural trend for each party, i.e., the species. In that sense, according to our results, the best approach to achieve satisfactory solutions is to use only the mutation and crossover operators, with mutation with a significantly higher frequency.

What is more, when compared to the previously implemented case-based approach, this bio-inspired model reveals some advantages. First of all, it is significantly more efficient. In fact, the case-based model requires several interactions with the database and some potentially time-consuming computational tasks (e.g. computation of similarity of each pre-selected case). Compared to this, the bio-inspired model is significantly more efficient, being able to generate more solutions in less time, mainly due to the simplicity of the genetic operators. In addition, two other disadvantages were identified in our case-based approach. On the one hand, in legal domains in which there were few cases, the system was unable to generate information with enough quality for the parties to take good decisions. On the other hand, changes in the law are quite frequent, which may render past cases useless. This is a disadvantage especially in civil law legal systems. Concerning this subject, the bio-inspired model has rewards when compared to the case-based one. In fact, the amount solutions being used is only limited

**Table 3.3**  
Final statistics of the run.

Genetic operators	Crossover	Heredity	Mutation	
	20	0	80	
Best overall solution	Generation: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.27 & 0.15 & 0.27 & 0.31 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	35	Mean fitness:	0.5250
Fitness of best solution for each species	Party1 0.7081200	Party2 0.7226	Party3 0.7226	Party4 0.7028

**Table 4.3**  
Final statistics of the run.

Genetic operators	Crossover	Heredity	Mutation	
	10	10	80	
Best overall solution	Generation: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0.37 & 0.17 & 0.11 & 0.35 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	65	Mean fitness:	0.5313
Fitness of best solution for each species	Party1 0.6785	Party2 0.7460	Party3 0.7460	Party4 0.7074

by the initial settings. This increases our degree of confidence in the computational process, once there will be always enough cases to be considered, which at the same time will cover virtually the entire search space, assuring variety. Moreover, changes in the laws of the legal systems will only imply a change in the rules embodied by the system, which is a relatively simple task.

## Acknowledgments

The work described here was developed under the TIARAC—Telematics and Artificial Intelligence in Alternative Conflict Resolution research project (PTDC/JUR/71354/2006), a research project supported by FCT (Science & Technology Foundation), Portugal. The work of Davide Carneiro was supported by a doctoral grant, also by FCT (SFRH/BD/64890/2009).

## References

- [1] H. Brown, A. Marriott, *ADR Principles and Practice*, Sweet and Maxwell, UK, 1999.
- [2] G. Peruginelli, G. Chiti, *Artificial Intelligence in Online Dispute Resolution*, Proceedings of the Workshop on the Law of Electronic Agents—LEA, 2002.
- [3] J. Sedano, L. Curiel, E. Corchado, E. Cal, J. Villar, A soft computing method for detecting lifetime building thermal insulation failures, *Integrated Comput. Aided Eng.* 17 (2) (2010) 103–115, IOS Press.
- [4] E. Corchado, A. Arroyo, V. Tricio, Soft computing models to identify typical meteorological days, *Logic Journal of the IGPL* (2010), Oxford University Press.
- [5] H. Raiffa, *The Art and Science of Negotiation*, Harvard University Press, 2002.
- [6] P.R.E. Walton, R.B. McKersie, *A Behavioral Theory of Labor Negotiations*, McGraw-Hill, 1965.
- [7] S.C. Bennett, *Arbitration: Essential Concepts*, ALM Publishing, 2002.
- [8] G. Chiti, G. Peruginelli, *Artificial Intelligence in Alternative Dispute Resolution*, Proceedings of LEA, pp. 97–104, 2002.
- [9] D.A. Waterman, J. Paul, M. Peterson, Expert systems for legal decision making, *Expert Syst.* 3 (1986) 212–226, <http://dx.doi.org/10.1111/j.1468-0394.1986.tb00203.x>.
- [10] F. Hayes-Roth, Rule-Based Systems, *Commun. ACM* 28, 9, 921–932, DOI=10.1145/4284.4286 <http://doi.acm.org/10.1145/4284.4286>, 1985.
- [11] J.L. Kolodner, *Case-Based Reasoning*, Morgan Kaufmann, San Mateo, CA, 1993.
- [12] D.A. Waterman, M.A. Peterson, *Models of Legal Decision-Making*, R-2717-ICJ, Institute for Civil Justice, Rand Corp., Santa Monica, CA, 1981.
- [13] L.T. McCarty, Reflections on TAXMAN: an experiment in artificial intelligence and legal reasoning, *Harvard Law Rev.* 90 (1977) 837–893.
- [14] D.A. Waterman, M. Peterson, Rule-Based Models of Legal Expertise, in: *Proceedings of the First National Conference on Artificial Intelligence*, Stanford University, 1980.
- [15] W.G. Popp, S. Judith, A computer program to advise lawyers in reasoning a case, *Jurimetrics J.* 5 (4) (1975) 303–314.
- [16] K. Atkinson, T. Bench-Capon, Legal case-based reasoning as practical reasoning, *Artif. Intell. Law* 13 (1) (2006) 93–131, <http://dx.doi.org/10.1007/s1050600690033>.
- [17] J.L. Kolodner, R.L. Simpson, The MEDIATOR: analysis of an early case-based problem solver, *Cognitive Sci.* 13 (1989) 507–549.
- [18] K. Sycara, in: D. Shapiro (Ed.), *The Persuader*. In *The Encyclopedia of Artificial Intelligence*, John Wiley and Sons Inc., New York, N.Y., 1992.
- [19] J. Popple, *A Pragmatic Legal Expert System*, Applied Legal Philosophy Series, Dartmouth (Ashgate), Aldershot, May, xviii + 384 pp., ISBN 1 85521 739 2, 1996.
- [20] F. Andrade, P. Novais, D. Carneiro, J. Zeleznikow, J. Neves, Using BATNAs and WATNAs in Online Dispute Resolution, In *New Frontiers in Artificial Intelligence*, Kumiyo Nakakoji, Yohei Murakami and Eric McCreedy (Eds.), JSAI-isAI 2009 Workshops, Revised Selected Papers, Springer—LNAI 6284, ISBN 978-3-642-14887-3, 2010, pp. 5–18.
- [21] D. Carneiro, P. Novais, F. Andrade, J. Neves, Retrieving Information in Online Dispute Resolution Platforms: A Hybrid Method, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Law, University of Pittsburgh School of Law, USA, ACM ISBN: 978-1-4503-0755-0, 2011, pp. 224–228.
- [22] D. Carneiro, P. Novais, F. Andrade, J. Neves, Improving Mediation Processes with Avoiding Parties, in Takashi Onada, Daisuke Bekki, Eric McCreedy (Eds.): *New Frontiers in Artificial Intelligence—JSAI-isAI 2010 Workshops*, LENLS, JURISIN, AMBN, ISS, Tokyo, Japan, November 18–19, 2010, Revised Selected Papers, Lecture Notes in Computer Science 6797, Springer, ISBN 978-3-642-25654-7, 2011, pp. 117–128.
- [23] E. Aarts, F. Grotenhuis, *Ambient intelligence 2.0: towards synergetic prosperity*, *J. Ambient Intell. Smart Environ.* 3 (2011) 3–11, IOS Press.
- [24] K. Thomas, R. Kilmann, *Conflict and Conflict Management*. Available at <http://www.kilmann.com/conflict.html>, 1974 (last accessed July 2011).
- [25] A. Abraham, Editorial—hybrid soft computing and applications, *Int. J. Comput. Intell. Appl.* 8 (1) (2009).
- [26] E. Corchado, A. Herrero, Neural visualization of network traffic data for intrusion detection, *Appl. Soft Comput.* 11 (2) (2011) 2042–2056.
- [27] D. Carneiro M. Gomes, P. Novais, F. Andrade, J. Neves, Automatic Classification of Personal Conflict Styles in Conflict Resolution, In *Legal Knowledge and Information Systems—JURIX 2011: The Twenty-Fourth Annual Conference*, Katie M. Atkinson (Ed), IOS Press—Frontiers in Artificial Intelligence and Applications, Vol. 235, ISBN 978-1-60750-980-6, 2011, pp. 43–52.

- [28] D. Carneiro, J.C. Castillo, P. Novais, A. Fernández-Caballero, J. Neves, M.T. López, Stress monitoring in conflict resolution situations, *Adv. Intell. Soft Comput.* 153, (2012) 137–144, [http://dx.doi.org/10.1007/978-3-642-28783-1\\_17](http://dx.doi.org/10.1007/978-3-642-28783-1_17).



**Davide Carneiro.** Researcher at the CCTC (Computer Science and Technology Center), in the Department of Informatics, University of Minho, Braga, Portugal. He is currently enrolled in a joint Doctoral Programme in Computer Science of three top Portuguese Universities. He develops scientific research in the field of Artificial Intelligence, namely Multi-agent Systems, Genetic Programming and Case-based Reasoning applied to the Legal field and Ambient Intelligence. He has participated in several research projects in the fields of Ambient Intelligence and Online Dispute Resolution. He is the co-author of several publications in his field of interest, including book chapters, journal papers and

conference and workshop papers.

He has been a member of the Program Committee of the 6th International Conference on Hybrid Artificial Intelligence Systems 2011 (HAIS 2011), member of the Scientific Committee of the International Symposium on Distributed Computing and Artificial Intelligence (DCAI 2011), Publicity and Web Chair of the Ambient Intelligence Environments (Aml Environments 2011) and Local Organization Committee of the International Symposium on Ambient Intelligence (ISAI 2010) and the International Workshop on Soft Computing Models in Industrial Applications (SOCO 2010).

In 2008 he has been awarded the TLeIA08—a National Award for Artificial Intelligence projects attributed by the Portuguese Artificial Intelligence Association and in 2009 he has been awarded an Academic Merit Scholarship by the Portuguese Government.



**Paulo Novais.** Professor of Computer Sciences at the Department of Informatics, in the University of Minho, Braga, Portugal and researcher at the CCTC (Computer Science and Technology Center). He received a PhD in Computer Sciences from the same university in 2003. He develops scientific research in the field of Artificial Intelligence, namely Knowledge Representation and Reasoning, Machine Learning and Multi-Agent Systems, with applications in the areas of Law and Ambient Intelligence mainly in Ambient Assisted Living.

In the last years, he has been the leader of the Intelligent Systems Lab (<http://islab.di.uminho.pt/>) at

the CCTC (Computer Science and Technology Center) in the University of Minho. He has led and participated in several research projects sponsored by Portuguese public and private Institutions and has supervised several PhD and MSc students. He is the co-author of over 100 book chapters, journal papers, conference and workshop papers and books.

He has been chair of the Organising or Scientific Committee of ISAI (International Symposium on Ambient Intelligence), SOCO 2011 (International Conference on Soft Computing Models in Industrial and Environmental Applications) and PROVE 2007 (IFIP Working Conference on Virtual Enterprises) and member of organising committee of several others events (e.g. Pervasive Health 2011, Intelligent Environments 2011, ECAI 2010).

He is Vice-president of APPIA, the Portuguese Association for Artificial Intelligence (2006–2011).



**José Neves.** He is full professor of computer science at Minho University, Portugal, since 1983. Jose Neves is the Deputy Director of the Division for Artificial Intelligence (AI). He received his PhD in Computer Science from Heriot Watt University, Scotland, in 1983. His current research interests relate to the areas of knowledge representation and reasoning and evolutionary intelligence, aiming to construct dynamic virtual worlds of complex symbolic entities that compete against one another in which fitness is judged by one criterion alone, intelligence, here measured in terms of a process of quantification of the quality of their knowledge, leading to the development of new procedures and other metaheuristics and their application in complex tasks of optimization and model inference in distinct areas, namely in the healthcare arena (e.g. machine learning in an intensive care unit environment, ambient intelligence).

and other metaheuristics and their application in complex tasks of optimization and model inference in distinct areas, namely in the healthcare arena (e.g. machine learning in an intensive care unit environment, ambient intelligence).