A novel game playing based approach to the modeling and support of consensus reaching in a group of agents

Maciej Świechowski*, Janusz Kacprzyk*, *Fellow of IEEE*, Sławomir Zadrożny*

*Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
Email: (m.swiechowski, kacprzyk, zadrozny)@ibspan.waw.pl

***Abstract*—The paper concerns the problem of reaching consensus among agents in group decision making. A popular framework of individual preferences expressed as (fuzzy) preference relations is adopted. The consensus reaching process is assumed to be based on a discussion in the group of agents, which is expected to make the initially expressed preferences closer one to another. We present a novel approach to the modeling of the consensus reaching process as a game, in the sense of game playing. We use the Monte Carlo Tree Search (MCTS) algorithm with the Upper Confidence Bounds Applied for Trees selection formula, which is a state of the art solution algorithm in that area. The consensus reaching process is modeled as a sequence of actions, referred to as moves, of the individual agents involved. A model of the assessment of a configuration of the individual preference relations is proposed. A decision support system that implements the approach proposed is developed, which provides the agents with an easy to read evaluation of the expected outcome of each move. The approach constitutes a new paradigm in the modeling of a consensus reaching process, and then its support.**

***Keywords*–group decision making, consensus reaching process support, game playing, rationality criteria**

I. Introduction

The problem of consensus reaching considered in this paper may be viewed as a follow up of the source problem of group decision making the essence of which can be summarized as follows. There is a (finite) group of agents (individuals, decision makers, ...) and a (finite) set of options (alternatives, variants, ...). Both these sets are assumed to be relatively small. The agents provide their testimonies as to the options, which are here assumed as the individual fuzzy (graded, with values from $[0, 1]$) preferences as to the particular pairs of options. The problem is to find a group decision solution (cf. Nurmi [1], [2]) meant as an option or a (maybe fuzzy) set of options that best represents the preferences of the whole group of agents. Such a solution, in the setting of fuzzy preferences, may be determined in various ways, notably by some dominance analyses (cf. Kacprzyk [3] or choice function analyses (cf., e.g. Kacprzyk and Zadrożny [4], [5], [6] and Świtalski [7]). Usually, these preferences differ in the beginning to a large extent and it is obvious that in such a case the group decision determined may be not meaningful enough.

Therefore, it may often have much sense to first perform the *consensus reaching process* to make the individual fuzzy preferences of the agents closer, and then to determine a group decision making solution. Of course, the precondition is that the agents are rationally committed to consensus, i.e., are ready to change their preferences.

We are concerned here with such a problem of consensus reaching which is considered a step by step process. Often, the moderator, a "superagent", is assumed who is responsible for running the consensus reaching session and for making the agents appropriately change their preferences, by using argumentation, suggesting mutual concessions, etc., to possibly increase the degree of consensus. The motivation of such a moderator driven consensus reaching process, and the architecture for its implementation through a computer based decision support system, is given, for instance, in Fedrizzi, Kacprzyk and Zadrożny [8]. The underlying concept of a soft degree of consensus due to Kacprzyk and Fedrizzi [9], [10], [11] is there employed in which a degree of consensus is basically meant as a degree to which, e.g., "*most* of the *relevant* (knowledgeable, expert, ...) agents agree as to *almost all* of the *important* options".

A number of approaches has been proposed to support such a consensus reaching process [8], [12], [13], [14], [15], [16]. They can be termed *data driven* in the sense that they only use data on agents and their preferences for deriving some additional information given as linguistic summaries.

In this paper, we pursue a novel approach. Namely, to derive an additional information on agents and their fuzzy preferences, we assume first that we employ a model of behavior of the group of agents (of course, in our case in the sense of changes of their preferences). More specifically, we assume that our consensus reaching session is represented as a game but not, as it is common in broadly perceived decision sciences, bargaining, negotiations, as a game in the sense of game theory, but in the sense of the so-called *general game playing* (GGP) [17] which basically boils down to the design of some tools and techniques that use elements of artificial intelligence to develop programs that can play various games which are represented by a set of logical expressions and rules. The game is then represented as a game tree and a sophisticated simulation framework is employed using the so-called Monte Carlo Tree Search (MCTS) algorithm. In such a way, we can find a rational way to determine the best sequence of changes of preferences. Therefore, since the game constitutes here a model of a *rational behavior* of the agents, its use implies a qualitatively new approach to the support of consensus reaching which can be termed a *model driven* approach.

We assume that the agents are rationally committed to the reaching of the consensus, i.e., they are ready to change their original preferences. On the other hand, they are assumed to also observe some other criteria. For example, it is reasonable to expect that while aiming at contributing to the consensus and being ready to change their preferences they would still like to *preserve their original original preferences* as much as possible. Moreover, they are also assumed to opt for a *fair treatment of preference of all involved agents*, i.e., that all agents would divert from their original preferences to a more or less the same extent.

Our goal in this paper is to propose a new general approach to modeling, and then supporting, the consensus reaching process based on a new model of a reasonable behavior of agents involved, in the sense of some well specified criteria for rational acts of the agents.

The model of an individual agent proposed comprises his or her appropriate multicriteria assessment of the quality of a given set of preference relations. The model of the consensus reaching process starts with our earlier model [14] and extends it with a new concept of a protocol (or, more generally, a family of protocols) defining some rules for some part of the discussion in the group. The latter concept is related to the (generalized) game playing and, accordingly, we propose to employ one of the most effective and efficient algorithms in this area, the MCTS, to carry out simulations of the discussion in the group of agents. The idea is to use its basic version as a generic platform for the simulations and then to model some approaches to the moderator driven discussion by using some specific variant of the generic implementation.

Section II briefly reminds a flexible measure of consensus proposed by Fedrizzi and Kacprzyk [9], [10] which is a core component of the proposed model of an agent involved. Then, other components/criteria of this model are introduced. In Section III, we formally introduce a novel model of the consensus reaching process and its support. In Section IV, the use of a chosen game-playing algorithm, called Monte Carlo Tree-Search, is discussed, which is the backbone of the support system. In the next section, we present the experimental setting and results. The last section is devoted to conclusions.

II. Measuring of a Degree of Consensus and a Model of an Individual

Formally, we consider the following setting for the core group decision making process. There is a set of $N \geq 2$ options, $S = \{s_1, s_2, ..., s_N\}$, and a set of $M \geq 2$ agents $E = \{e_1, e_2, ..., e_M\}$. This model may be further extended considering, e.g., relevance degrees of the options and importance weights of the agents. In the simplest case, each agent $e_m \in E$ expresses his or her preferences in the form of an individual *fuzzy preference relation* R_m in $S \times S$. Thus R_m is a fuzzy set in $S \times S$ and its membership function $\mu_{R_m}(s_i, s_j)$ may be given such an interpretation that $\mu_{R_m}(s_i, s_j) > 0.5$ denotes the preference degree of option s_i over option s_j as expressed by agent e_m, and it is understood that the higher this degree the stronger this preference. On the other hand, $\mu_{R_m}(s_i, s_j) < 0.5$ denotes the preference degree of option s_j over option s_i and it is understood that the lower this degree the stronger this preference. Finally, $\mu_{R_m}(s_i, s_j) = 0.5$ denotes the indifference between options s_i and s_j. Usually, the reciprocity of the relation R_m is assumed, i.e., $\mu_{R_m}(s_i, s_j) + \mu_{R_m}(s_j, s_i) = 1$ and thus, the membership function $\mu_{R_m}(s_i, s_j)$ has to be specified for $i < j$ only.

A. Measuring the degree of consensus

The traditional understanding of the consensus as an unanimous agreement is not applicable for this scenario: one cannot expect the total agreement regarding preferences on all pairs of options under consideration, and by all agents. Moreover, such a total agreement is usually not needed cf., e.g., Loewer and Laddaga [18]. Therefore, a new definition of the consensus was proposed [9], [8], [10] which is based on the concept of *fuzzy (soft) majority*. A natural manifestations of such a "soft" majority are the so-called *linguistic quantifiers* as, e.g., "most", "almost all", "much more than a half", etc. Such linguistic quantifiers can be dealt with by, e.g., fuzzy logic based calculi of linguistically quantified statements as proposed by Zadeh [19].

The new degree of consensus proposed in [8] can be equal to 1, which stands for full consensus, when, e.g., "*most of the agents agree as to almost all relevant options*". The particular elements of this definition of consensus are modeled using fuzzy logic concepts. The relevance of opitions is assumed to be given as a fuzzy set B defined in the set of options S such that $\mu_B(s_i) \in [0,1]$ is a *degree of relevance* of option s_i, from 0 for fully irrelevant to 1 for fully relevant, through all intermediate values. The relevance b_{ij} of a pair of options, $(s_i, s_j) \in S \times S$, may be defined as:

$$b_{ij}^B = \frac{1}{2}[\mu_B(s_i) + \mu_B(s_j)] \tag{1}$$

which is clearly the most straightforward option; evidently, $b_{ij}^B = b_{ji}^B$, and b_{ii}^B do not matter; for each i, j.

Then, the degree of consensus is derived as follows:

First, for each pair of agents (e_m, e_n) and each pair of options (s_i, s_j) a degree of agreement $\nu_{ij}(m,n)$ is derived; it is, in general, computed as a function $Aggrm$ of two arguments $(\mu_{R_m}(s_i, s_j), \mu_{R_n}(s_i, s_j))$ and can take different forms provided that $Aggrm(x,x) = 1$ and $Aggrm(0,1) = Aggrm(1,0) = 0$ and the monotonicity of $Aggrm(x,y)$ with respect to $|x - y|$ is preserved. For example,

$$\nu_{ij}(m,n) = \begin{cases} 1 & \text{if} \quad \mu_{R_m}(s_i, s_j) = \mu_{R_n}(s_i, s_j) \\ 0 & \qquad \text{otherwise} \end{cases}$$

Second, for each pair of agents (e_m, e_n) a degree of agreement $\nu_{Q_1}^B(m,n)$ as to their preferences between Q_1 (a linguistic quantifier as, e.g., "most", "almost all", "much more than 50%", etc.) pairs of relevant options is derived as:

$$\nu_{Q_1}^B(m,n) = \mu_{Q_1}\left(\frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}[\nu_{ij}(m,n) \wedge b_{ij}^B]}{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} b_{ij}^B}\right) \tag{2}$$

where $\wedge$ is a t-norm operator, e.g., the minimum.

Third, these degrees are aggregated to obtain a degree of agreement $con(Q_1, Q_1, I, B)$ of Q_2 (a linguistic quantifier similar to Q_1) pairs of important agents as to their preferences between Q_1 pairs of relevant options, and this is meant to be the *degree of consensus* sought:

$$consensus = \mu_{Q_2}\left(\frac{\sum_{m=1}^{M-1}\sum_{n=m+1}^{M}[\nu_{Q_1}^B(m,n) \wedge b_{mn}^I]}{\sum_{m=1}^{M-1}\sum_{n=m+1}^{M} b_{mn}^I}\right) \tag{3}$$

B. A model of an individual agent rationally committed to consensus

We assume that the agents are rationally committed to reaching the consensus. Hence, they are assumed to discuss various aspects of the decision problem under consideration and then are willing and ready to change their initially expressed preferences so as to get closer to the opinions of the other members of the group. Thus, in order to model their behavior we assume that their decision with this respect are guided by their assessment of the quality of the current *state* (form) of the preference relations of all involved agents. A crucial component of such a quality assessment (via an objective function) is a measure of the degree of consensus, such as the one given by (3). However, an individual may be assumed to take into account also other aspects of the current state of the preferences and in the new model proposed in this paper we include the following aspects:

- the *consensus* degree (3),
- the *cost*; the difference (distance) between the initial preferences of an agent and those at a current state of the process,
- the *fairness* of the current state meant as the similarity of the distances of the initial preferences of all agents to their preferences at the current state
- the *bias* towards the initial opinions (preferences) of the majority of agents meant as tendency to limit the changes in preferences to a small number of agents.

We introduce the objective function, meant as a crucial element of our model of the rationally committed (to consensus) agent, as a linear combination of these objectives:

$$objective = w_1 * consensus + w_2 * (1 - cost) + \quad (4)$$
$$+ w_3 * fairness + w_4 * bias$$

The *degree of consensus* component is calculated as previously shown in (3).

The *cost* (of change) component is defined as a degree measuring how much the preferences of agent e_m in the state for which the objective function is calculated differ from the desired preferences. There can be various realizations of this concept. In this paper, we define the *cost* as:

$$cost = 1 - \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\left| \mu_{R_m}(s_i, s_j) - \mu'_{R_m}(s_i, s_j) \right| \wedge b_{ij}^B \right)}{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} b_{ij}^B} \quad (5)$$

where:

- $\mu_{R_m}(s_i, s_j)$ - is the current (possibly changed in the process) preference degree regarding a pair of options (s_i, s_j) as expressed by the individual e_m;
- $\mu'_{R_m}(s_i, s_j)$ - is the initial value of this preference degree

The value of *fairness* is calculated as:

$$fairness = 1 - \sigma(cost_i) \quad (6)$$

where: $\sigma(costs_i)$ is the standard deviation of the agents's *costs*. The higher the value of *fairness*, the more similar is the extent to which all agents have to revise their initial preferences to reach the current state. Thus, all agents may be expected to be equally committed to the emerging group solution.

The *bias* is calculated as:

$$bias = \text{median}\{1 - cost_i\}_{i=1,\ldots,M} \quad (7)$$

If, for example, the value of *bias* is equal to 0.8, then it means that *at least a half* of agents have diverted only *at most* to the extent 0.2 from their initial preferences to reach the current state. The maximization of this bias measure, combined with the maximization of the consensus degree, results in preferring such situations, in which if there is initially a high degree of consensus among a half of more agents, then they should not be forced to change their preferences to a large extent to reach consensus in the group.

It is worth noting that the degrees of consensus, fairness and bias are global quality indicators in the sense that they take the same value for all agents at a given state. On the other hand, the cost component takes different values for different agents. Moreover, there is obviously an interplay between components of the objective function (4). For example, the fairness and bias may be viewed to be in conflict but the idea of introducing the bias is the following. If in the initial state of the preferences there is a relatively high degree of consensus in the group, then we would prefer it to be a seed for a further steps of the consensus reaching process, i.e., we would like not to force those agents who lag "behind" the consensus to change their opinions, even at the expense of the value of the fairness indicator. On the other, if the degree of consensus is low at the initial state, then this component will not play an important role in guiding the behavior of an individual agent.

The weights in (4) may be set separately for each individual or may be set globally for all of them. A variety of ways to set the weights of the function gives us a testing framework for various models of the rational behavior. For example, we can verify the impact of particular settings of weights on the final outcome of the consensus reaching process.

The objective function implicitly models the rational behavior of the individual agents, i.e., we assume that they will generally try to change their preferences in such a way that maximizes the objective function. Moreover, we propose to use this model to support the consensus reaching process and suggest to the agents to change their opinions according to this model, as it is described in the next section.

III. A NEW APPROACH FOR THE SUPPORT OF CONSENSUS REACHING

Now, we will elaborate on details of a novel model of the consensus reaching process in our setting. The most important element of the model is the concept of a *move* of an agent. A move of an individual agent consists in choosing by him or her a pair of options and a degree of preference for them, and inviting other agents to accept such a degree of preference. In the basic setting to be assumed in this paper, this degree will be identical with the degree of preference for this pair of options originally revealed by a given agent. It may be assumed that particular agents, in the course of the discussion in the group, will propose those pairs of options which are the most important for them. In a more general setting, each agent may propose all pairs of options for consideration.

Formally, each move is a 3-tuple: $(s_i, s_j, \mu_{R_m}(s_i, s_j))$, where the respective parameters are the two options being compared and a preference degree stating how much option s_i is preferred to s_j

The process starts with the preparation of the collection of moves from each agent. The moves are submitted to the system, i.e., the algorithms employed in the supporting computer program have access to all of them. The way and moment of process initialization, as well as the choice of the sequence of agents whose moves are subsequently considered, depend on a particular scenario and agenda adopted. The approach proposed is however general and can be tuned to a particular case.

The consensus reaching process is iterative, and in each iteration after some discussion, an agent, who is currently active, may *play*, i.e., make a move. The other agents may then *accept* or *reject* it. The acceptance implies changes in the current preference of the agent who has accepted a move regarding the particular pair of options that move has concerned. Then, the next agent (player) becomes active in a round robin fashion. The process ends when a desired degree of consensus, set in advance, has been reached or all agents have used all their possible moves.

As stated above, the agents (players) have two types of decisions to make:

- *Play* (make one move) from the set of all moves. The action is available to the currently active agent.
- *Accept* or *Reject* a move made by the active player.

Before making each decision, the support system, which is based on computer simulations, will evaluate each option in terms of its expected quality. The expected quality is calculated according to the objective function which has been introduced and discussed in Section II-A. This function assigns to each *state of the game*, i.e., the set of preference relations of all agents, a number from $[0, 1]$ where 0 and 1 denote the lowest and the highest outcomes, respectively. The outcome is calculated for each agent individually just like the scores in games. The idea is to simulate lots of possible future scenarios, in an intelligent way, similarly to how it is performed for the game tree search in games in order to suggest the best move to an agent. The details of the simulation based algorithm to be employed in this work will be presented in Section IV.

The underlying idea is based on the following facts:

- Some aspects of the problem can be modeled as a game so that we claim that game playing algorithms may provide strong tools and techniques to find the best decision in the current state of the game.
- We present a new paradigm, which is suitable for situations where the agents would prefer to check first "what can happen" if they propose a certain preference to others for acceptance/rejection or if they accept/reject a preference proposed by another agent.
- The approach is based on the rationale that the participating agents will gain trust [20] in the computer system and its supporting evaluation.
- There is no in depth analysis of moves by the system such as the breaking them down into arguments and hence using elements of argumentation theory. Therefore, the approach is conceptually clear and yet relatively general and flexible. For example, one could construct a setup with a different set of possible moves and the rest of the process could remain the same.

A general architecture of the consensus reaching process is illustrated in Figure 1. The process is gamified, i.e., represented as a game. However, we have to distinguish between the current game running through discussion and the simulated games performed by computer simulation. The current game is only just one in the process considered (because it is the process itself), whereas there are lots of simulated games in each iteration. In Figure 1, the steps present only in the current game are highlighted in a light blue color. All other steps are common for both types of games.

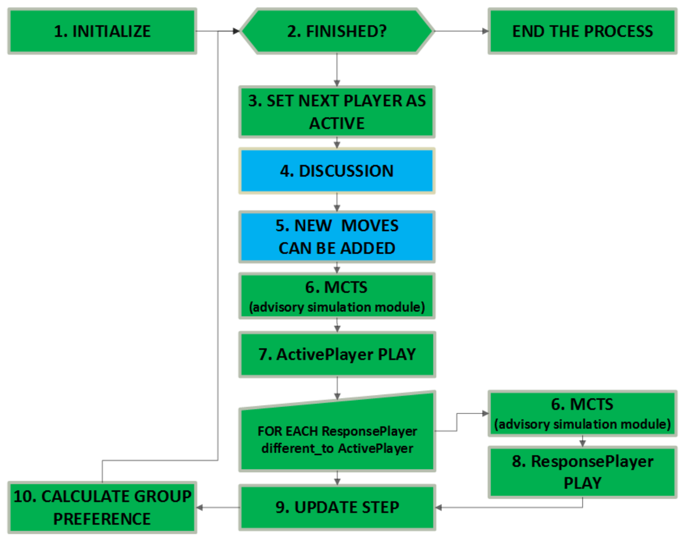

Figure 1. The overall structure of the consensus reaching support system

Below we will comment on the steps shown in Figure 1:

1) **Initialize** - the agents submit their moves to the system. For each agent, the system will manage a dynamic collection of his or her moves.
2) **STOP?** - the meeting is over if one of the following events occurs: (1) a desired threshold of a consensus degree is reached, (2) the session exceeds a certain allotted time slot, or (3) the agents have no more yet unpresented moves left.
3) **Set next agent as active** - the next agent (player) becomes active to openly make a move. Assuming that the agents' indices are kept from 1 to M, such a circular loop can be attained by the following updating:

   $$ActivePlayer := [(ActivePlayer+1)\%(M+1)]+1$$

   where M is the total number of agents.
4) **Discussion** - the discussion is held according to the agenda adopted.
5) **New moves can be added** - after the discussion, the agents can submit new moves to the system or override the previous ones. If they do, the MCTS module will reset the construction of the game tree to take all the changes into account.

6) **MCTS (advisory simulation module)** - the simulation based advisory module is realized by the Monte Carlo Tree Search (MCTS) algorithm. The input to the algorithm is the complete current state (the preference matrices as well as the active agent's index). Section IV is solely devoted to the algorithm, so we will not go into details here.
7) **ActivePlayer PLAY** - this is the active agent's (player's) turn to make a move ("play"). He or she can choose a move from their earlier submitted set of moves or pass if they do not have any one left.
8) **ResponsePlayer PLAY** - if the active agent (player) has made (played) a move, then each other player can make a move ("play") by accepting or rejecting it. If the active agent quits, then this step is skipped.
9) **Update Step** - a move made is then considered as being used and it is removed from the set of agent's moves. If an agent accepts a move $(s_i, s_j, \mu_{R_m}(s_i, s_j))$, then an update of other agents' current preferences to the pair of options s_i, s_j to the value of $\mu_{R_m}(s_i, s_j)$ occurs. Such an action also removes all moves the agent had regarding the same pair of options.
10) **Calculate Group Preference** - the current degree of consensus is calculated.

IV. MCTS BASED DECISION SUPPORT

Let P be an iterative decision problem in which each decision may lead to a different state (problem configuration). The Monte Carlo Tree Search (MCTS) [21] is an algorithm of searching a tree composed of states that represent the nodes and decisions that represent the edges. The most notable application of the MCTS is in the game playing domain [22], [23], [24], [25], where the problem configurations are game states while the decisions are actions in the game which players can make.

The MCTS works by performing massive amounts of computer simulations of the problem and gather statistical evidence about the available actions by growing a tree of the game. A single iteration of the algorithm consists of four phases as depicted in Figure 2.

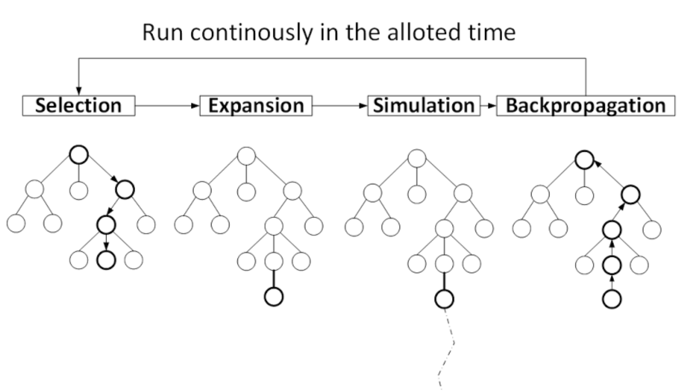

Figure 2. Four phases which comprise the MCTS algorithm.

1) **Selection.** In this phase, the algorithm starts from the root node and traverses down the tree by choosing subsequent children nodes. The child node at each node down the path is chosen according to the so-called selection policy. The selection phase ends when there is no child node to choose, i.e., a leaf node has been reached.
2) **Expansion.** One of the possible actions is applied to a node reached in the previous step and the tree is grown by adding a child node representing the resulting state.
3) **Simulation.** The algorithm starts from the new node and performs a complete game simulation, i.e., reaching a terminal state. This phase is done outside the game tree and no nodes are added. Once the simulation reaches the terminal state, the obtained goals (outcomes) of each player are checked. In this work, the goals are calculated using the objective function (c.f. Section II-A and Equation 4). The terminal state is defined as having at least one of the two properties: (1) a desired threshold of a consensus degree is reached, (2) the agents have no more yet unpresented moves left.
4) **Back-propagation.** Here, the statistics are recalculated inside all nodes along the path from the root to the leaf (containing the starting state for the simulation) in the game tree. The statistics include the average scores of each player and the number of visits to a node. An average score is computed as the total score attained in the iterations going through a particular node divided by the number of visits to that node.

If a terminal node is added in the current iteration then the current iteration of the MCTS algorithm proceeds directly to the back propagation phase.

In the classic implementation, the simulation phase is performed randomly. However, the selection phase, which searches over the already constructed part of the tree, is based on a more informed search that balances the exploration (of less explored branches) and exploitation (of the most promising moves done so far). The widely used algorithm to implement this idea, which we adopt in the paper, is called the Upper Confidence Bounds applied for Trees (UCT) [26], [27].

$$a^* = \arg \max_{a \in A(s)} \left\{ Q(s,a) + C \sqrt{\frac{ln\,[N(s)]}{N(s,a)}} \right\} \quad (8)$$

where $A(s)$ is a set of actions available in state s, $Q(s,a)$ denotes the average result of playing action a in state s in the simulations performed so far, $N(s)$ is the number of times the state s has been visited in previous simulations and $N(s,a)$ is the number of times the action a has been sampled in this state in previous simulations. Constant C controls the balance between exploration and exploitation. In this paper, the set $A(s)$ depends on the fact whether the current player is the active player to make a move or not. In the former case, the available actions are moves which have not yet been played by the player. In the latter case, the available actions are *accept* and *reject* the move.

In this work, we use the most common value of $C = \sqrt{2}$, suggested in the literature [28], which makes the second term in (8) equal to the size of the one-sided confidence interval for the average reward using the Chernoff-Hoeffding bounds. The

MCTS algorithm equipped with the UCT formula is proven to converge to the game theoretic optimum given sufficient time [26]. The authors of [26] also show that the failure probability of selecting a suboptimal action at the root of the tree (where the current decision is to be made) converges to zero at a polynomial rate with respect to the number of simulations.

If the theoretical convergence is not required, the UCT formula can be replaced by the so-called UCB-1-Tuned formula [29] which often performs more efficiently in practice.

In summary, the input to the algorithm is the current state of the problem. This state is stored in the root in the tree searched. The state in our problem contains: the list of options (alternatives) as to which the consensus is sought, the preferences of each player (agent) with respect to these options, the moves of the players and who is the active player. Then, the method will gradually search the tree, i.e., the space of possible options, through many iterations. Each iteration will add a node to the tree. Actions within the tree are chosen according to (8) whereas actions outside the tree (in the simulation phase) are chosen randomly. The participants assisted by the MCTS are presented with the expected scores Q for each action available to them at the moment.

V. Empirical Results

In general, in order to evaluate the efficacy of a consensus-reaching support software, it is required to ask human experts to carry out a discussion while using this software to enrich the process. However, such an experiment has several flaws, e.g., it is not very repeatable under the same conditions and also human decisions can be affected by their current mood or disposition. The system proposed in this paper has a unique feature that it advises to take actions based on the results coming from the MCTS algorithm, therefore, just for the purpose of testing, the human participants are not required. We decided to perform a lot of automated experiments using three test scenarios with the assumption that an action which is advised by the system is always chosen (hence, no human intervention is required). The biggest disadvantage of having no participants is that no new moves will be made on the fly during the experiments. We only provide the machine agents with the initial set of moves and they can propose them as well as accept or reject moves made by other agents. The process is illustrated in Figure 1 with the blue steps excluded, because they require human involvement. The experiments are aimed at revealing the emerging nature of the system, in particular, the way in which the MCTS-based advisor suggests actions and the consequences of following these advises. Naturally, in a real-world setting, the participants do not need to agree with the advisor, but this is always the case when people and not automated systems are responsible for making the choice, ultimately.

A. Experimental Setup

We have prepared base test scenarios of three decision processes. The first one assumes that there are 3 participants and 5 options. The preferences regarding each pair of options greatly differ amongst the experts (all 0s vs. all 0.5s vs. all 1s), so it is very difficult to reach a consensus in such a situation. The next two setups employ 6 participants and 3 options. In the former one, there is a majority group of 4 experts who have common preferences, so there is a consensus within the group, but they are vastly different than preferences of people outside this group. In the latter case, the situation is flipped over, i.e., there is a minority group of 2 experts with a consensus on start and the preferences of the 4 remaining ones differ significantly.

For each test scenario, we have conducted 81 experiments varying over the parameter settings. The following values of the objective function parameters (i.e., weights w_1-w_4 in Equation 4) were tested: $consensus \in \{0.8, 0.6, 0.4\}$, $cost \in \{0.6, 0.4, 0.2\}$, $fairness \in \{0.6, 0.3, 0.0\}$, $bias \in \{0.6, 0.3, 0.0\}$. The $consensus$ range contains slightly higher values, because this is the main objective in the problem, so more often than not it should have the biggest weight assigned. The $fairness$ and $bias$ ranges include zero, because these two measures are not as natural as the previous two and can be regarded as optional. Our aim was to test how the system behaves with and without them.

For the clarity sake, we present only selected, more interesting, results of the 81 experiments per each test.

We will provide some comments to Table I, which contains weights for the respective measures and their values after executing the consensus-reaching process. Row 1 has the lowest consensus value for all experiments with consensus weight equal to 0.8. Rows 1-3 present how significantly, the weight of $bias$ affects the resulting consensus degree. Rows 4-5 show how setting a lesser weight for $fairness$ slightly increases the consensus value in this setting. Row 6 compared to rows 7-9 shows that in this test, higher $fairness$ prevents reaching a full (1.00) consensus. Rows 7-9 are examples of a full consensus. However, rows 10 shows that setting each weight to 0.6 results in a total conservative situation, where players do not change their preferences at all. Indeed, the weight set to reaching the consensus is outweighed by the remaining criteria. Rows 11-15 show how lowering certain weights increases the consensus value. However, a significant increase is observed only starting from Row 16 to Row 18. We can also see in rows 16-18 that the second expert can retain their intial preferences the most, because they were inbetween the preferences of the other two experts. Then, lowering the consensus weight in Rows 19-22 leads to a complete disagreement (Consensus = 0.00). This can again be mitigated by lowering the $Cost$ weight as shown in Rows 22-25, but the maximum value of consensus reachable with these weight is only 0.73. In general, we observe that:

- The ulility function is the most efficiently maximized for the weights of *consensus*, *cost*, *fairness* and *bias* equal to $0.8, 0.2, 0$ and 0, respectively.
- The higher the weight of consensus, the higher the value of consensus.
- The higher the weight of cost, the lower the value of consensus.
- Combination of both high *cost* and *bias* prevents the system from reaching a consensus
- Each parameter has an impact on the results and the dependence does not have a simple linear nature.

The second test with results shown in Table II is relatively easy for reaching the consensus. In most of the cases, the preferences of the majority group of experts who agree with themselves are retained. However, it is worth taking a closer look at rows 24 and 27. Here, the combination of $bias$ weight

TABLE I. This table contains results of the experiment conducted for the first test scenario. The first column shows an ordinal numbers of experiments with particular parameters setting. The next four columns show weights assigned to respective criteria: *consensus*, *cost*, *fairness* and *bias*. The bolded column shows the final consensus value obtained after running the program, and two subsequent columns show the same for for *fairness* and *bias*, while the next one shows individual costs values for each player. The cost value of 1.00 means that the player did not change their preferences at all. The last column denotes the final value of the objective function, cf. (4)

.

Row	Weights:				Values:				
	cons	cost	fair.	bias	cons	fair.	bias	costs	objective
1	0.8	0.6	0.3	0.6	**0.53**	0.98	0.80	0.75, 0.80, 0.80	0.73
2	0.8	0.6	0.3	0.3	**0.80**	0.82	0.70	0.45, 0.90, 0.70	0.75
3	0.8	0.6	0.3	0.0	**1.00**	0.78	0.55	0.45, 0.95, 0.55	0.84
4	0.8	0.4	0.3	0.3	**0.67**	1.00	0.75	0.75, 0.75, 0.75	0.76
5	0.8	0.4	0.0	0.3	**0.73**	0.95	0.75	0.75, 0.75, 0.65	0.73
6	0.8	0.2	0.6	0.3	**0.93**	0.92	0.60	0.60, 0.70, 0.50	0.84
7	0.8	0.2	0.3	0.3	**1.00**	0.86	0.65	0.65, 0.65, 0.35	0.85
8	0.8	0.2	0.0	0.6	**1.00**	0.76	0.75	0.25, 0.75, 0.75	0.85
9	0.8	0.2	0.0	0.0	**1.00**	0.91	0.50	0.50, 0.70, 0.50	0.91
10	0.6	0.6	0.6	0.6	**0.00**	1.00	1.00	1.00, 1.00, 1.00	0.75
11	0.6	0.6	0.3	0.3	**0.13**	1.00	0.95	0.95, 0.95, 0.95	0.69
12	0.6	0.6	0.0	0.0	**0.50**	0.85	0.85	0.60, 0.95, 0.85	0.65
13	0.6	0.4	0.6	0.6	**0.57**	0.98	0.80	0.75, 0.80, 0.80	0.78
14	0.6	0.4	0.6	0.0	**0.73**	1.00	0.70	0.70, 0.70, 0.70	0.82
15	0.6	0.4	0.0	0.6	**0.57**	0.76	0.90	0.40, 0.90, 0.90	0.73
16	0.6	0.4	0.0	0.0	**1.00**	0.76	0.50	0.50, 1.00, 0.50	0.87
17	0.6	0.2	0.3	0.0	**1.00**	0.92	0.55	0.45, 0.65, 0.55	0.90
18	0.6	0.2	0.0	0.0	**1.00**	0.88	0.60	0.60, 0.70, 0.40	0.89
19	0.4	0.6	0.6	0.6	**0.00**	1.00	1.00	1.00, 1.00, 1.00	0.82
20	0.4	0.6	0.0	0.0	**0.00**	1.00	1.00	1.00, 1.00, 1.00	0.60
21	0.4	0.4	0.6	0.3	**0.00**	1.00	1.00	1.00, 1.00, 1.00	0.76
22	0.4	0.4	0.3	0.3	**0.00**	1.00	1.00	1.00, 1.00, 1.00	0.71
23	0.4	0.2	0.6	0.0	**0.73**	1.00	0.70	0.70, 0.70, 0.70	0.86
24	0.4	0.2	0.0	0.6	**0.73**	0.86	0.80	0.80, 0.80, 0.50	0.76
25	0.4	0.2	0.0	0.0	**0.67**	1.00	0.75	0.75, 0.75, 0.75	0.70

equal to 0 and $fairness$ weight equal to 0.6 was enough to "break" the majority group at least a bit. These two examples are settings for which the consensus was reached in a more fair way regarding the minority group.

Finally, Table III presents the results for the experiment that involved two experts with the same preferences and four with significantly different ones. Depending on the weights assigned, a full consensus may be found when the other four experts adjust their preferences to be the same as the first two experts' preferences (rows 9 or 17). However, it can also be reached when the first pair of experts changes their preferences to some extent (e.g. row 7). The lowest value of consensus is achieved when both *cost* and *fairness* weights are set high, so the players refrain from changing their preferences as possible (row 20). However, a non-zero weight of consensus also induces some changes, what can been seen by looking at costs of experts 4 and 6 in row 20.

VI. Conclusions

We have proposed a new paradigm of the consensus reaching process support system. Its main feature is the submit-and-test nature. The participants in the process first submit their testimonies to the system. In the background, a machine-learning-based module performs simulations and evaluates the efficacy of available actions by means of their expected outcome at the end of the process. The outcome is calculated according to the objective (goal) function based on four components: consensus, cost, fairness and majority-cost-based (*bias*) degrees. The function is our attempt to provide a set of criteria modelling relatively rational agents' behavior when it is simulated by the machine.

TABLE II. This table contains results of the experiment conducted for the second test scenario. Please refer to Table I for the interpretation of the values.

Row	Weights:				Values:				
	cons	cost	fair.	bias	cons	fair.	bias	costs	objective
1	0.8	0.6	0.6	0.6	**0.80**	0.94	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.93	0.92
2	0.8	0.6	0.3	0.6	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.95
3	0.8	0.6	0.3	0.3	**0.80**	0.94	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.93	0.90
4	0.8	0.6	0.0	0.6	**0.89**	0.90	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.77	0.94
5	0.8	0.6	0.0	0.0	**0.80**	0.94	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.93	0.87
6	0.8	0.4	0.6	0.6	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.94
7	0.8	0.4	0.6	0.0	**1.00**	0.95	0.83	1.00, 1.00, 1.00, 1.00, 0.75, 0.93	0.95
8	0.8	0.4	0.3	0.6	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.96
9	0.8	0.4	0.3	0.3	**0.89**	0.90	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.77	0.92
10	0.8	0.2	0.6	0.6	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.95
11	0.8	0.2	0.6	0.3	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.94
12	0.8	0.2	0.6	0.0	**1.00**	0.96	0.75	1.00, 1.00, 1.00, 1.00, 0.83, 0.85	0.96
13	0.8	0.2	0.0	0.0	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.98
14	0.6	0.6	0.6	0.6	**0.80**	0.94	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.93	0.93
15	0.6	0.6	0.6	0.0	**1.00**	0.95	0.83	1.00, 1.00, 1.00, 1.00, 0.75, 0.93	0.93
16	0.6	0.4	0.6	0.6	**0.80**	0.94	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.93	0.92
17	0.6	0.4	0.6	0.3	**0.80**	0.94	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.93	0.91
18	0.6	0.4	0.6	0.0	**1.00**	0.95	0.83	1.00, 1.00, 1.00, 1.00, 0.75, 0.93	0.94
19	0.6	0.4	0.3	0.0	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.93
20	0.6	0.4	0.0	0.0	**0.89**	0.90	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.77	0.91
21	0.6	0.2	0.6	0.6	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.94
22	0.6	0.2	0.0	0.0	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.97
23	0.4	0.6	0.6	0.6	**0.80**	0.94	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.93	0.94
24	0.4	0.4	0.6	0.0	**1.00**	0.95	0.83	0.83, 0.83, 0.83, 0.83, 0.75, 0.93	0.93
25	0.4	0.4	0.3	0.6	**0.82**	0.93	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.85	0.93
26	0.4	0.4	0.3	0.3	**0.80**	0.94	1.00	1.00, 1.00, 1.00, 1.00, 0.83, 0.93	0.92
27	0.4	0.2	0.6	0.0	**1.00**	0.98	0.75	0.75, 0.75, 0.75, 0.75, 0.83, 0.85	0.94
28	0.4	0.2	0.0	0.0	**1.00**	0.84	1.00	1.00, 1.00, 1.00, 1.00, 0.58, 0.77	0.96

TABLE III. This table contains results of the experiment conducted for the third test scenario. Please refer to Table I for the interpretation of the values.

Row	Weights:				Values:				
	cons	cost	fair.	bias	cons	fair.	bias	costs	objective
1	0.8	0.6	0.6	0.6	**1.00**	0.87	0.83	0.83, 0.83, 0.53, 0.75, 0.93, 0.68	0.88
2	0.8	0.6	0.6	0.3	**0.89**	0.90	0.85	0.92, 0.92, 0.7, 0.67, 0.85, 0.77	0.87
3	0.8	0.6	0.6	0.0	**0.89**	0.89	0.85	0.92, 0.92, 0.65, 0.67, 0.85, 0.77	0.86
4	0.8	0.6	0.0	0.0	**0.89**	0.84	0.77	1.00, 1.00, 0.7, 0.58, 0.77, 0.68	0.85
5	0.8	0.4	0.0	0.6	**0.89**	0.90	0.85	0.92, 0.92, 0.7, 0.67, 0.85, 0.77	0.86
6	0.8	0.4	0.0	0.3	**1.00**	0.84	0.85	0.92, 0.92, 0.45, 0.67, 0.85, 0.77	0.91
7	0.8	0.2	0.3	0.0	**1.00**	0.90	0.75	0.75, 0.75, 0.62, 0.83, 0.85, 0.6	0.94
8	0.8	0.2	0.0	0.6	**1.00**	0.76	0.82	0.87, 0.87, 0.23, 0.45, 0.63, 0.82	0.89
9	0.8	0.2	0.0	0.0	**1.00**	0.78	0.77	1.00, 1.00, 0.37, 0.58, 0.77, 0.68	0.95
10	0.6	0.6	0.6	0.6	**0.69**	0.89	0.93	1.00, 1.00, 0.7, 0.83, 0.93, 0.77	0.85
11	0.6	0.6	0.6	0.0	**0.89**	0.89	0.85	0.92, 0.92, 0.65, 0.67, 0.85, 0.77	0.86
12	0.6	0.6	0.3	0.6	**0.58**	0.92	0.93	1.00, 1.00, 0.9, 0.83, 0.93, 0.77	0.82
13	0.6	0.4	0.3	0.0	**1.00**	0.84	0.85	0.92, 0.92, 0.45, 0.67, 0.85, 0.77	0.89
14	0.6	0.3	0.0	0.6	**0.80**	0.83	0.93	1.00, 1.00, 0.7, 0.58, 0.93, 0.68	0.85
15	0.6	0.4	0.0	0.0	**0.89**	0.90	0.85	0.92, 0.92, 0.7, 0.67, 0.85, 0.77	0.86
16	0.6	0.2	0.6	0.6	**1.00**	0.87	0.83	0.83, 0.83, 0.53, 0.75, 0.93, 0.68	0.89
17	0.6	0.2	0.0	0.0	**1.00**	0.78	0.77	1.00, 1.00, 0.37, 0.58, 0.77, 0.68	0.93
18	0.4	0.6	0.6	0.6	**0.60**	0.91	0.93	1.00, 1.00, 0.82, 0.83, 0.93, 0.77	0.85
19	0.4	0.6	0.6	0.3	**0.64**	0.88	0.93	1.00, 1.00, 0.9, 0.75, 0.93, 0.68	0.84
20	0.4	0.6	0.6	0.0	**0.56**	0.90	0.93	1.00, 1.00, 0.9, 0.75, 0.93, 0.78	0.81
21	0.4	0.4	0.6	0.6	**0.62**	0.94	0.92	0.92, 0.92, 0.9, 0.83, 0.93, 0.77	0.86
22	0.4	0.4	0.6	0.3	**0.71**	0.93	0.90	0.92, 0.92, 0.9, 0.75, 0.85, 0.77	0.85
23	0.4	0.4	0.6	0.0	**0.89**	0.92	0.93	0.83, 0.83, 0.73, 0.75, 0.93, 0.68	0.87
24	0.4	0.4	0.0	0.6	**0.58**	0.92	0.93	1.00, 1.00, 0.9, 0.83, 0.93, 0.77	0.82
25	0.4	0.4	0.0	0.3	**0.71**	0.93	0.90	0.92, 0.92, 0.9, 0.75, 0.85, 0.77	0.81
26	0.4	0.4	0.0	0.0	**0.71**	0.93	0.90	0.92, 0.92, 0.9, 0.75, 0.85, 0.77	0.78
27	0.4	0.2	0.0	0.6	**0.80**	0.89	0.92	0.92, 0.92, 0.7, 0.67, 0.93, 0.77	0.86
28	0.4	0.2	0.0	0.3	**1.00**	0.84	0.85	0.92, 0.92, 0.45, 0.67, 0.85, 0.77	0.90
29	0.4	0.2	0.0	0.0	**1.00**	0.84	0.85	0.92, 0.92, 0.45, 0.67, 0.85, 0.77	0.92

The objective value is calculated as a linear combination of those criteria and the weights can be manipulated to better suit a particular problem. The function is approximated with the help of the Monte Carlo Tree Search (MCTS) algorithm, which has proven itself to be very succesful in predicting the outcome in games. The participants actually play a "virtual game of consensus". They can play their testimonies, which other players may in turn accept or reject.

The help of the machine advisor, running in the background, may be used in a variety of ways. The most natural is to increase the confidence of participants of proposing certain testimonies and arguments related to them. It is always a human who is responsible for making the choice, but they may take advantage of the machine suggestion. The system can also help to determine which preferences are the most imporant from the perspective of reaching the consensus.

Another way to use the system is to use it as a platform for the simulation of the group of agents involved in the consensus reaching process. Assuming that the agents behavior is guided by the maximization of (4), this behavior may be simulated using the MCTS machinery and on top of that some other algorithms to suggest the actions to an individual agent may be tested.

In order to evaluate the system, we prepared three distinct scenarios in which a consensus is sought. Then, we assumed that each participant will agree with the machine advisor to show the consequences of this behaviour. The results show that the system will try to fulfill each expert's individual goal within the constraints imposed by the given objective function. The objective function can be fine-tuned to meet specific needs.

References

[1] N. H, "Approaches to collective decision making with fuzzy preference relations," Fuzzy Sets and Systems, vol. 6, no. 3, 1981, pp. 249–259.

[2] H. Nurmi, "Fuzzy social choice: a selective retrospect," Soft Comput., vol. 12, no. 3, 2008, pp. 281–288.

[3] J. Kacprzyk, "Group decision making with a fuzzy majority via linguistic quantifiers. Part I, Part II," Cybernetics and Systems: An International Journal, vol. 16, 1985, pp. 119–129,131–144.

[4] J. Kacprzyk and S. Zadrożny, "Computing with words in decision making through individual and collective linguistic choice rules," International Journal of Uncertainty Fuzziness and Knowledge Based Systems, vol. 9, 2001, pp. 89–102.

[5] ——, "Collective choice rules in group decision making under fuzzy preferences and fuzzy majority: a unified OWA operator based approach," Control and Cybernetics, vol. 31, no. 4, 2002, pp. 937–948.

[6] ——, "Towards a general and unified characterization of individual and collective choice functions under fuzzy and nonfuzzy preferences and majority via the ordered weighted average operators," Int. J. Intell. Syst., vol. 24, no. 1, 2009, pp. 4–26.

[7] Z. Świtalski, "Choice functions associated with fuzzy preference relations," in Non-Conventional Preference Relations in Decision Making, J. Kacprzyk and M. Roubens, Eds. Berlin: Springer-Verlag, 1988, pp. 106–118.

[8] M. Fedrizzi, J. Kacprzyk, and S. Zadrożny, "An interactive multi-user decision support system for consensus reaching processes using fuzzy logic with linguistic quantifiers," Decision Support Systems, vol. 4, no. 3, 1988, pp. 313–327.

[9] J. Kacprzyk and M. Fedrizzi, "Soft consensus measures for monitoring real consensus reaching processes under fuzzy preferences," Control and Cybernetics, vol. 15, no. 3-4, 1986, pp. 309–323.

[10] ——, "A 'Human-consistent' Degree of Consensus Based on Fuzzy Logic With Linguistic Quantifiers," Mathematical Social Sciences, vol. 18, no. 3, 1989, pp. 275–290.

[11] ——, "A'soft' measure of consensus in the setting of partial (fuzzy) preferences," European Journal of Operational Research, vol. 34, no. 3, 1988, pp. 315–325.

[12] E. Herrera-Viedma, F. Herrera, and F. Chiclana, "A Consensus Model for Multiperson Decision Making With Different Preference Structures," IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 32, no. 3, 2002, pp. 394–402.

[13] J. Kacprzyk, Z. S, and R. Z, "How to support consensus reaching using action rules: anovel approach," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 18, no. 4, 2010, pp. 451–470.

[14] J. Kacprzyk and S. Zadrożny, "Soft computing and web intelligence for supporting consensus reaching," Soft Computing, vol. 14, no. 8, 2010, pp. 833–846.

[15] E. Herrera-Viedma, F. J. Cabrerizo, J. Kacprzyk, and W. Pedrycz, "A review of soft consensus models in a fuzzy environment," Information Fusion, vol. 17, 2014, pp. 4 – 13.

[16] I. Palomares and L. Martínez-López, "A semisupervised multiagent system model to support consensus-reaching processes," IEEE Trans. Fuzzy Systems, vol. 22, no. 4, 2014, pp. 762–777.

[17] M. R. Genesereth, N. Love, and B. Pell, "General Game Playing: Overview of the AAAI Competition," AI Magazine, vol. 26, no. 2, 2005, pp. 62–72.

[18] B. Loewer and R. Laddaga, "Destroying the consensus," Synthese, vol. 62, no. 1, 1985, pp. 79–95.

[19] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages," Computers & Mathematics with applications, vol. 9, no. 1, 1983, pp. 149–184.

[20] J. Lee and N. Moray, "Trust, control strategies and allocation of function in human-machine systems," Ergonomics, vol. 35, no. 10, 1992, pp. 1243–1270.

[21] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," IEEE Transactions on Computational Intelligence and AI in Games, vol. 4, no. 1, 2012, pp. 1–43.

[22] M. Świechowski and J. Mańdziuk, "Self-Adaptation of Playing Strategies in General Game Playing," IEEE Transactions on Computational Intelligence and AI in Games, vol. 6, no. 4, Dec 2014, pp. 367–381.

[23] S. Gelly et al., "The Grand Challenge of Computer Go: Monte Carlo Tree Search and Extensions," Commun. ACM, vol. 55, no. 3, Mar. 2012, pp. 106–113.

[24] O. Syed and A. Syed, Arimaa - A New Game Designed to be Difficult for Computers. Institute for Knowledge and Agent Technology, 2003, vol. 26, no. 2.

[25] G. Chaslot, S. Bakkes, I. Szita, and P. Spronck, "Monte-Carlo Tree Search: A New Framework for Game AI." in Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, 2008, pp. 216–217.

[26] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning," in Proceedings of the 17th European conference on Machine Learning, ser. ECML'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 282–293.

[27] L. Kocsis, C. Szepesvári, and J. Willemson, "Improved Monte-Carlo Search," Univ. Tartu, Estonia, Tech. Rep, vol. 1, 2006.

[28] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," Machine learning, vol. 47, no. 2-3, 2002, pp. 235–256.

[29] S. Gelly and Y. Wang, "Exploration exploitation in Go: UCT for Monte-Carlo Go," in NIPS: Neural Information Processing Systems Conference On-line trading of Exploration and Exploitation Workshop, Canada, Dec. 2006.