Knowledge-Based Systems 34 (2012) 64–80

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Genetic fuzzy markup language for game of NoGo

Chang-Shing Lee [a,*], Mei-Hui Wang [a], Yu-Jen Chen [a], Hani Hagras [b], Meng-Jhen Wu [a], Olivier Teytaud [a,c]

[a] *Department of Computer Science and Information Engineering, National University of Tainan, Taiwan*
[b] *The Computational Intelligence Centre, School of Computer Science and Electronic Engineering, University of Essex, UK*
[c] *TAO (Inria), LRI, UMR, CNRS, 8623, bat 490 Univ. Paris-Sud, F-91405 Orsay, Cedex, France*

ARTICLE INFO

Article history:
Available online 21 February 2012

Keywords:
Fuzzy ontology
Fuzzy markup language
Monte-Carlo Tree Search
Game of Go
NoGo

ABSTRACT

NoGo is similar to the game of Go in terms of gameplay; however, the goal is different: the first player who either suicides or kills a group loses the game and the first player with no legal move loses the game. In this paper, we propose an approach combining the technologies of ontologies, evolutionary computation, fuzzy logic, and fuzzy markup language (FML) with a genetic algorithm (GA)-based system for the NoGo game. Based on the collected patterns and the pre-constructed *fuzzy NoGo ontology*, the genetic FML (GFML) with the fuzzy inference mechanism is able to analyze the situation of the current game board and then play next move to an inferred good-move position. Additionally, the *genetic learning mechanism* continuously evolves the adopted GFMLs to enable an increase in the winning rate of the GA-based NoGo via playing with the baseline NoGo. In the proposed approach, first, the domain experts construct the important NoGo patterns and the *fuzzy NoGo ontology* based on the rules of NoGo and the past game records. Second, each GA-based NoGo as *White* plays against the baseline NoGo as *Black* according to the inferred and calculated good-move position, respectively. Third, the *genetic learning mechanism* is carried out to generate two new evolved GFMLs and then the worst two GFMLs stored in the GFML repository are replaced. Fourth, the GFML with the highest winning rate is randomly sampled from the GFML repository in the time series. Finally, one by one the GA-based NoGo adopts the sampled GFML to play lots of games against the baseline NoGo to obtain the winning rate of the GA-based NoGo. The acquired winning rates at the time series show that the proposed approach can work effectively and that the average winning rate of the GA-based NoGo program is much stronger than the baseline NoGo program.

1. Introduction

After Deep Blue's famous victory over Kasparov in 1996, some of the research focus shifted to games, where alpha–beta search was not sufficient. Most prominent among these games was the ancient Asian game of Go [24]. Game of Go is one of the main challenges in artificial intelligence. In particular, it is much harder than chess in spite of the fact that it is fully observable and has very intuitive rules. Computer Go has been developing for the past several years. In 1990, Abramson [1] proposed to model the expected outcome of a game by averaging the results of many random games. In 1993, Brugmann [7] proposed Monte-Carlo techniques for Go using almost random games, and developed the refinement. Ten years later, Bouzy's Indigo program used Monte-Carlo simulations to decide between the top moves proposed by a classical knowledge-based Go engine [6]. Today, programs such as MoGo/MoGoTW, Crazy Stone, Fuego, Many Faces of Go, and Zen have achieved a level of play that seemed unthinkable only a decade ago. These programs are now competitive at a professional level for 9×9 Go and amateur Dan strength on 19×19 [24,25,36,32].

Ontology has been extensively studied in many research fields. Additionally, it has been widely pointed out that the classical ontology is not sufficient to deal with imprecise and vague knowledge for some real-world applications. On the other hand, the fuzzy ontology can effectively help to handle and process uncertain data and knowledge [9,22,29]. The fuzzy ontology is an extension of the domain ontology that is more suitable to describe the domain knowledge for solving the uncertainty reasoning problems [26]. For instance, Lee et al. [22,23] proposed a type-2 fuzzy ontology to apply to personal diabetic-diet recommendation and diet assessment. They also proposed a fuzzy ontology to apply to news summarization [26] and diabetes decision support [21]. Yager and Petry [38] developed a multicriteria approach to data summarization using concept ontologies and a framework for the resolution of apparently contradictory evidence for decision making. Buche et al. [8] designed a fuzzy querying scheme for incomplete, imprecise, and heterogeneously structured data in the relational model using ontologies and rules. Bechhofer et al. [5] used ontologies and vocabularies for dynamically linking to solve some problems with static, restricted, and inflexible traditional web.

* Corresponding author.
E-mail addresses: leecs@mail.nutn.edu.tw (C.-S. Lee), hani@essex.ac.uk (H. Hagras), olivier.teytaud@lri.fr (O. Teytaud).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2012.02.006

Fuzzy Markup Language (FML) is a software technology to create fuzzy oriented abstraction tools. FML is also a fuzzy-based markup language that can manage fuzzy concepts, fuzzy rules, and a fuzzy inference engine. FML is composed of an eXtensible Markup Language (XML), a document-type definition, and an extensible stylesheet language transformations and its syntax is realized by using a set of tags representing the different components of a fuzzy controller [2,3,29]. For the past years, the Ontology Application and Software Engineering (OASE) Lab. of National University of Tainan (NUTN), Taiwan has applied FML to many research topics [23,35,37]. Additionally, LASA Lab. of University of Salerno, Italy proposed an Ambient Intelligence (AmI) fuzzy computing system by using the FML technology to model the inferred services and implemented AmI framework on heterogeneous hardware [3]. Turowski and Weng [34] proposed an XML-based approach to represent and process fuzzy information to better integrate business applications into company-wide business application systems.

Based on the FML, a genetic fuzzy markup language (GFML) is developed in this paper to manage fuzzy concepts, fuzzy rules, and a fuzzy inference engine, thanks to genetic learning. Genetic algorithm (GA) is inspired by the mechanism of natural selection, where stronger individuals are likely the winners in a competing environment. Through the genetic evolution method, an optimal solution can be found and represented by the final winner of the genetic game [30]. Rao and Pratihar [31] used GA to design an optimal knowledge base to develop a fuzzy logic (FL)-based expert system to predict the results of finite element analysis. Hadavandi et al. [17] presented an integrated approach based on genetic fuzzy systems and artificial neural networks for constructing a stock price forecasting expert system. In this paper, we combine the genetic algorithms (GAs) with the FML to form a *GFML-based fuzzy learning mechanism*, and apply the GFML to the game of NoGo domain. Based on the proposed *fuzzy NoGo ontology*, the developed approach is able to optimize the membership functions and rules.

Playing the game of Go is a very personalized behavior because each player has his/her own thinking even for the identical situation. In addition, there are variable strategies to choose from the current game board, which makes the game of Go very challenging for the artificial intelligence. Alam et al. [4] proposed the possibility measure algorithm and the averaging conflict aggregation algorithm to handle the difficulty of the conflict aggregation process in multi criteria decision making. Cebi and Kahraman [10] developed a decision support system based on fuzzy information axiom to help decision maker to solve their decision problems. Liang and Wang [28], Tang and Chang [33], Fenton and Wang [16], Kelemenis and Aksounis [19], Chang [11], and Yue [39] proposed the related method to decision making to apply to decision problems. Nevertheless, how to make the computer Go programs memorize as many patterns as humans, the ontology techniques might be one of the solutions. Hoock et al. [18] presented the techniques of the ontologies for learning agents in Monte-Carlo Tree Search (MCTS) and experimented it in the game of Go. Coulom [14], Chaslot et al. [12], Kocsis and Szepesvari [20], and Rimmel et al. [32] defined the main concepts in MCTS and one of the most well-known variants is Upper Confidence Bounds (UCBs) applied to trees. This tree will be biased in order to explore more deeply moves that have good results so far, and this is done by the repetition of four steps, namely, descent, evaluation, update, and growth.

This paper also applies a hybrid approach employing ontologies, genetic algorithms (GAs), and GFML to the game of NoGo, where the acquired winning rates at the time series show that the proposed approach can work effectively and that the average winning rate of the GA-based NoGo is much higher than our baseline NoGo implementation. The currently strongest approach for NoGo is Monte-Carlo Tree Search [13]; in particular, Fuego won the Birs 2011 competition just by adapting Fuego's code to the NoGo case. Other approaches include alpha–beta search with counting of legal moves and (non tree search) Monte-Carlo simulations. We here work on a winning possibility module, which is exactly a Monte-Carlo algorithm. Our goal is to show how a genetic algorithm can improve a GFML-based framework so that it improves a given program; the experiments are performed on this Monte-Carlo setting, but the same methodology could be applied to other approaches as well. This remainder of this paper is organized as follows. Section 2 introduces the game of NoGo. The GA-based game of NoGo is described in Section 3. The *GFML-based fuzzy learning mechanism* is introduced in Section 4. Section 5 shows the experimental results. Finally, the conclusions and future work are shown in Section 6.

2. Game of NoGo

In this section, the game of NoGo and its rules are introduced, and then some patterns are given to discuss the tactics of playing NoGo. Finally, the *fuzzy NoGo ontology* is depicted.

2.1. Introduction to NoGo

The NoGo game has syntactic similarity with the Go game in the sense that both have the notion of group, killing, black and white stones placed alternately on the board, and stones do not move. The gameplay is the same. However, the goal of NoGo is different from the one of Go; whereas, the winner in Go is determined by territory considerations at the end of the game, in NoGo, the loser is the first player who either (1) captures a group, (2) suicides, or (3) passes or resigns. As a consequence, NoGo is not at all tactically related to Go and it has been invented by the organizers of the Birs workshop on Combinatorial Game Theory 2011 for being a completely new game [13]. The rules of NoGo are as follows: (1) the

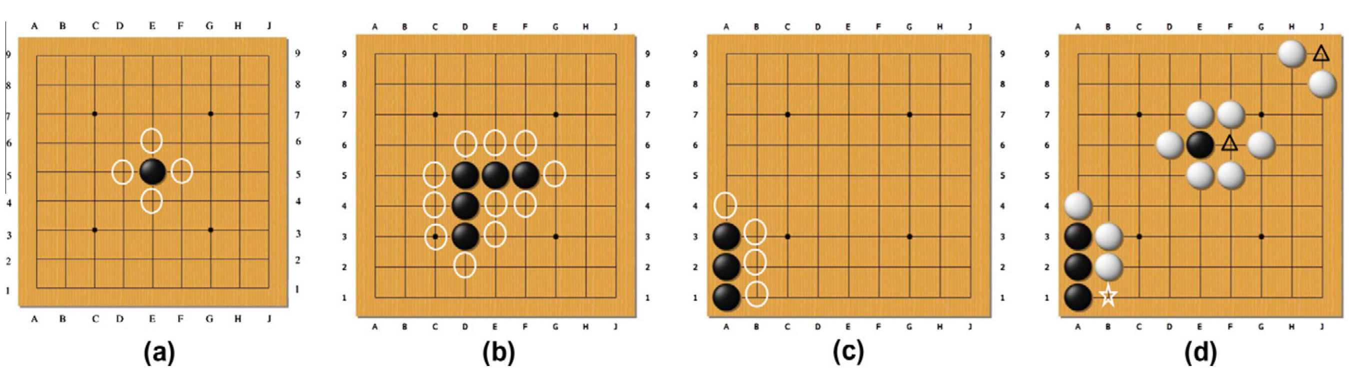

Fig. 1. Descriptions of the NoGo rules, (a)–(c) if all of the circle-marked positions are occupied by white stones, then the black group is dead, and (d) illegal moves for NoGo (J9) is illegal for *Black*, B1 is illegal for *White*, and F6 is illegal for both *Black* and *White*.