Social Computing Article[*]

Tabajara, Lucas
Institute of Informatics
Federal University of Rio Grande do Sul
Brazil
lmtabajara@inf.ufrgs.br and Prates, Marcelo
Institute of Informatics
Federal University of Rio Grande do Sul
Brazil
morprates@inf.ufrgs.br

Abstract

Abstract

1 Introduction

For years multi-agent systems have been used to research cooperation as a tool for problem-solving. Recently, however, there has been an increasing interest in the study of human beings as problem-solving agents. Several experiments have been conducted in which subjects are connected in a network with the goal of collectively solving a specific problem, and those have helped shed some light on the way humans interact to solve problems. However, those studies are limited to obtaining observations on human behaviour and formulating hypotheses, being unable to explain with certainty that behaviour. In this paper, we use a multi-agent based simulation to complement the study of human computation, as a way of explaining the strategies used by humans and understanding their consequences in a cooperative environment.

Human beings are known to be able to easily perform tasks which are still generally difficult for computers, such as natural language processing and image recognition. However, it would be useful to be able to apply human computation to more classical computer problems with precise definitions and algorithms, but which are still computationally intensive. The different sets of abilities between computers and humans suggest that human computation might provide a new approach to those problems, one that computers can't easily apply. However, we still don't know the limits of human abilities in problem-solving and how they compare to more traditional techniques. To take full advantage of human problem-solving skills, we must learn their limitations. In order to obtain that knowledge, we have developed a multi-agent system that simulates human behaviour according to findings of experiments performed in humans.

2 Background

The experiments conducted by [Farenzena *et al.*, 2011] had human beings trying to solve constraint satisfaction problems, namely Boolean Satisfiability (SAT) and the popular *Sudoku* game, with individuals connected through the network being able to exchange partial solutions of the problem in question. The behaviour observed in that study has provided evidence that human beings in a cooperative problem-solving environment might not evaluate solutions proposed by their peers, instead choosing the most readily available one.

Another result of those observations is that there is a higher probability of individuals copying peers' solutions when those are shared by several neighbours, referred to by the authors as *conformist behaviour*. A similar behaviour was observed in [Mason and Watts, 2011] for a different problem.

3 Contribution

We have built a multi-agent system that simulates the experiments of [Farenzena *et al.*, 2011], using the model of human behaviour proposed by them. We adapted that behaviour into a set of rules which we modeled in the *Memetic Network* model proposed by [Lamb and Araujo, 2008]. For simplicity, we limited our experiments to the Sudoku problem. Our goal is verifying whether the strategies employed by humans are competitive with other heuristics. After identifying the points in which those are lacking, we will then propose methods to increase the efficiency of human problem-solving networks.

4 System Overview

We developed a system in which the environment of the *Sudoku* problem-solving social network presented in [Farenzena *et al.*, 2011] could be modelled through autonomous agent networks and analysed through a series of experiments. The optimization technique of *Memetic Networks* [Lamb and Araujo, 2008] was employed on the modelling, providing a basis for the dynamics of the agent network.

5 Sudoku Solving Memetic Network

A *Memetic Network Algorithm* is composed of an ordered set of N agents, each encoding a complete solution to the optimization problem, and a binary $N \times N$ matrix representing possible connections between agents. Additionally, a *Memetic Network Algorithm* is composed of a set of rules specifying how connections between agents are formed and erased and how interactions between these agents take place. These rules are grouped under the categories of *Connection Rules*, specifying how agents will connect and disconnect

[*] This title is temporary

from each other; *Aggregation Rules*, specifying the dynamics of the information flow through connections; and *Appropriation Rules*, specifying how agents are supposed to add local changes to the information received through their connections [Lamb and Araujo, 2008].

Our solution adapts the *Memetic Newtork* technique - originally intended for use in optimization problems - to the context of constraint satisfaction problems. In this specific case (*Sudoku* solving), we treat problems with one unique solution. To deal with this scenario, we propose a *Memetic Network* variation in which each agent encodes not a complete, but a partial solution to the problem.

Our solution employs a fixed topology for the agent network and models cooperation through a set of *aggregation rules* and reasoning through a set of *appropriation rules*. Our aggregation rules specify how *Sudoku* partial solutions are copied from agent to agent, while our appropriation rules specify how agents increment these solutions with *Sudoku* solving techniques such as those studied in [Davis, 2007].

5.1 Aggregation Rules

The experiments conducted by [Farenzena *et al.*, 2011] point out a series of observations about the dynamics of cooperation in problem solving with human beings. For instance, the authors analysis has shown that users tend to copy not the most complete solutions, but, instead, the first solutions on the graphic interface. Additionally, we know from this same paper that users rarely copy solutions from more than 6 neighbours. In order to analyse and compare this behaviours, we developed a collection of aggregation rules, each modelling a copying strategy.

Pick Most Filled
This rule employs the intuitive strategy of copying from the neighbour with the most filled *Sudoku* board, namely the most complete solution.

Pick Among First
We know that users tend to copy the first (from left to right) solutions on the graphic interface [Farenzena *et al.*, 2011]. The authors have provided us with a mathematical model of this behaviour, stated as $\langle X(k) \rangle = (1-p)^{k-1}p$, where the parameter p is fixed as $p = 0.5479$ and $\langle X(k) \rangle$ denotes the probability of an agent copying the k_{th} neighbour solution.

We inserted this behaviour into our model by firstly generating a random ordering of neighbours for each agent in order to compose a simulated graphic interface. As a result, each agent visualizes some neighbours before or after others. Secondly, we translated the above mathematical model into an aggregation rule in which the solution copied by an agent is the k_{th} with probability $X(k)$.

5.2 Appropriation Rules

In *Sudoku* strategy literature, as in chess, we find multiple techniques with iconic names. Some popular examples are the *naked singles* rule, the *hidden singles* rule and the *naked twins* rule, studied in [Davis, 2007]. These strategies intend to, given a *Sudoku* puzzle in a partial state of completion, generate movements to mark blank cells of the puzzle, as *"mark cell 1 of column 3 with value 5"*. We reproduced a variety of these strategies, modelling each one of them as a function that maps a *Sudoku* partial solution to a set of movements.

Only Choice
This technique marks a *Sudoku* cell with a value only if this cell is the last blank cell on its row, column or 3 × 3 block.

Naked Singles
This rule marks a *Sudoku* cell with value v only if v is the single possible value to mark that cell. The other values must have been eliminated through a process of verifying that they are all present in the span encompassing the cell's row, column and 3 × 3 block.

Hidden Singles
This rule is similar to the *naked singles* rule. It is able to mark cells with more than one candidate value by looking for cells that are the only cells in their row, column or 3 × 3 block that can hold a particular value.

Two out of Three
This rule applies to groups of three contiguous 3 × 3 blocks. It aims to find a value v such that v is present in two of the three 3 × 3 blocks encompassed by the group, but missing on the third. It proceeds by enumerating all the candidate cells - namely the empty cells - on this block, and then, by eliminating from this set all the cells that are encompassed by the rows or columns in which v is placed on the other two blocks. If the resulting set has cardinality 1, the rule has successfully found one cell to mark.

Naked Twins

6 Results

Results

7 Conclusions

Conclusions

References

[Davis, 2007] Tom Davis. The mathematics of sudoku. November 2007.

[Farenzena *et al.*, 2011] Daniel Farenzena, Ricardo Araujo, and Luis Lamb. Collaboration emergence in social networks with informational natural selection. In *2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing*, 2011.

[Lamb and Araujo, 2008] Luis Lamb and Ricardo Araujo. Memetic networks: analyzing the effects of network properties in multi-agent performance. In *AAAI'08*, pages 3–8, 2008.

[Mason and Watts, 2011] Winter Mason and Duncan J. Watts. Collaborative learning in networks. November 2011.