A new software application for Backgammon based on a Heuristic Algorithm

MANUELA PANOIU, IONEL MUSCALAGIU, CAIUS PANOIU, RAZVAN VASILIU
Electrical Engineering and Industrial Informatics Department
Polytechnic University of Timisoara
Revolutiei Street, no 5 Hunedoara, cod 331128
ROMANIA
{m.panoiu, mionel, c.panoiu }@fih.upt.ro

Abstract: this paper presents a new software package based on an intelligent algorithm for playing backgammon. The software is implemented in Java language. The software allows both a game between two human players and between a human player and computer. In case of playing versus computer, a heuristic algorithm was used. The software package contains a module for tracking several backgammon games in the local area network.

Key-Words: Java, games, heuristic algorithm, artificial intelligence

1. Introduction
The complexity of computer games has increased over the last years. Games have attracted many researchers in artificial intelligence [1]. Games have always been an important application area for heuristic algorithms. There are several types of games: games that use perfect information and games that use imperfect information. There are also deterministic games (Chess, Go, Othello) and games with chance (Backgammon, Bridge, Scrabble, Poker). Board games such as checkers and backgammon have used intelligent techniques in the past decades. Because of working with imperfect information and because of chance a good algorithm for playing backgammon is difficult to implement.

The first computer software for backgammon was BKG 9.8. It was programmed by Hans Berliner in the late 1970s on a PDP-10 as an experiment in evaluating board positions. Early versions of BKG played badly even against poor players [4]. Beginning in the late 1980s, backgammon-playing software began to have more success with a neural network [1], [2] and [3].

In this paper we apply heuristic algorithm strategies for playing versus computer. The software package include two components: a component that can be used for playing backgammon can be used to play between two human players or between a human player and a computer; the second component was designed for monitoring several backgammon games in the local area network.

2. Backgammon – a game of chance
Backgammon is a nondeterministic game with chance. In nondeterministic games, chance is introduced by dice or card-shuffling. In case of Backgammon the chance is introduces by dice. Playing backgammon is simple: there are two players with 15 pieces each. The final goal is to move all pieces off the board. The rules are: Dice roll determines number of moves, players move in opposite directions, pieces cannot put on a point occupied by 2 or more of opponent’s pieces, single piece can be “hit” if it is put on the same line a piece of the opponent and hit piece must start a new route. The backgammon board is shown in figure 1.

There are some typical algorithms used for implement computer programs for playing backgammon. Such an algorithm is min max algorithm [5]. Minimax is a generalization of Alpha-Beta search for minimax trees with chance nodes. [5]. In case of games with chance a variation of min-max algorithm can be used. Such an algorithm is expectiminimax. In this algorithm, the outcome depends of a combination of the player's skill and chance elements such as dice [6]. The expectiminimax tree is a specialized variation of a minimax game tree that plays two-player zero-sum games (such as backgammon). In addition to "min" and "max" nodes of the traditional minimax tree, this variant has "chance" nodes, which take the expected value of a random event occurring [6], [7]. The form of the game tree for backgammon is shown in fig. 2 [7]:
3. The software package

The software package was implemented in Java programming language, because of the advantages of this language: is simple, platform-independent, distributed, secured and robust. Several classes have been implemented for modeling the players, the table game, the game, the moves, a ToolBar class, a class for the left and right panel, etc.

3.1. The “Game” application

The “Game” class is used for playing a game between two human players or between a human player and the computer. This class is a public class, extended from JFrame class and was designed to allow, through a user-friendly graphical user interface, to play a game. The GUI is shown in fig. 3. From a drop-down menu, the user can select some options for playing. The GUI allows starting a new game, to save a game or to load a game, as is shown in fig. 4a. A new game can be started between two human players or between a human player and the computer program. This option can be selected as is shown in figure 4b. Other options of the software can be selected as is shown in fig. 4c and fig. 4d.

The difficulty level can be selected only if it has previously been set to play against the computer, like in fig 4d. There are three levels of difficulty. For the first level were not use intelligent techniques. The program simply picks a random move from the list of all legal moves and performs the move. This option of the software can be used for the beginner’s players to learn to play backgammon. The second level uses a heuristic algorithm. For doing this it was calculated and assigned to all legal moves a numeric value called “merit factor”. Then all the legal moves was sorted by this merit factor and stored in a linked list. The computer will choose always the legal move with the maximum merit factor. For this level wasn’t considered the possibility of removing the opponent's pieces from the board. The third level uses also a merit factor for all legal moves at the time. The difference from the second level is that in this case the merit factor is higher if the player (the computer) can “hit” an opponent’s piece and remove it from the board.

To modeling a player was designed an implement the “Player” class. This class describe a player: the name of the player, a logical variable (indicates if the player can move on the next move or not), the number of pieces out of the player’s home, the number of pieces out of the game, the number of removed pieces, and a matrix for the dices. Another variable of the Player class indicates the pieces color: white or black.

For representing a move for a piece the board the lines from the board were numbered as is shown in figure 1. So the position of a single piece is described using the line number and the color of the piece. In order to be able to save all the legal moves were designed and implement the follow classes: a class name “MoveDices”. This class contained a matrix that describes a move: the position of the moved piece, and the dice values.
Fig. 3 The Graphical User Interface for playing backgammon

a) Load, save or begin a new game

b) Selecting the players

c) Other options

d) Selecting the difficulty level

Figure 4. The main options for the „Game”
The matrix has four lines and three columns. The last two lines can be used in case of double dice moves. The first two lines keep a simple dice move. For moving one piece it was store: the color of the piece, the line of start and the value of the dice. There are cases when it is possible to move a single piece. For these cases it was also designed and implements a class MoveADice. The class is a simplified version of the class MoveDice.

3.2. The “server” application
This class was implemented for monitoring several games in the local area network. If players want to play in the network first at all the server application must be started.

The “Server” class contain a ServerSocket variable and a port number for performed client server socket communication. In figure 5 is show the graphical user interface for this application.

For starting the monitoring the user must select from the “Administration” menu the “Start” option. Then, the players shall introduce the IP Address for the opponent player and the port number for communication with the server application. Then will check the “Connect” option from the “Options” menu as is shown in fig. 6.

The communication between the two applications can be made by using a “Client” class. This class contain a Socket variable for connect to the Server application
The “Config” menu can be used for port number configuration for communication with the client application and for displaying a log file with a history of the connections.

4. Conclusions
Our application of new software for playing backgammon is intuitive and simple to use. It can be used for many purposes: by the beginners to learn to play backgammon; to play between two players in the network and to play versus computer. In the future we intend to improve the algorithm for playing versus computer using new intelligent techniques.

References