Combining Deliberation and Reactive Behavior for AI Players in the Mini-Tichu Card-game

Martha Vlachou-Konchylaki
University of Patras
Patras, Greece
vlachoumartha@gmail.com

Stavros Vassos
Sapienza University of Rome
Rome, Italy
vassos@dis.uniroma1.it

ABSTRACT
Tichu is a popular trick-taking climbing card game that involves strong strategic play along with elements of chance. Apart from the typical search-based methods that employ domain-dependent or domain-independent mechanisms for selecting the most promising action at each step of the game, the complex dynamics that arise from the team-play feature of Tichu suggests the use of agent-based techniques for specifying a higher-level tactical strategy for artificial intelligent players. In this work we focus on finite state machines, a technique that has been traditionally used for specifying the behavior of non-player characters in video-games, and investigate how they can be coupled with simple search-based methods as a means for developing strong AI players in a simplified version of the game that we call Mini-Tichu.

Keywords
Artificial Intelligence, Card Games

1. INTRODUCTION
In order to illustrate the different roles for the search methods and the finite state machines, we investigate a slightly simplified version of the Chinese card game of Tichu, that we call Mini-Tichu. This game is played with a normal 52-card poker deck by four people that form two teams such that teammates sit in alternating positions. The deck is entirely dealt between the players and the objective of the game for each player is to discard all their cards. The first who achieves this gets a certain bonus, and one of the most important strategical points of the game is to plan for being first in this respect or help your teammate achieve this, while making it difficult for opponents to do the same thing. Mini-Tichu is a trick taking climbing game in the sense that each player can play by discarding a higher combination of cards from his hand. The player starting a trick defines the type of combination played, including a single card and a pair of cards, as well as all poker combinations, and a few more. Players may decide if they want to play a combination of the same type and higher value or pass. The last player that discarded a combination after three players have passed wins the trick and defines a new one.

Mini-Tichu is an incomplete information game as each player starts by knowing only his cards (1/4 of the full deck). Each round is fully observable as each action reveals hidden cards that are removed from the game. Also, the game is highly dynamic since there is a strong team-play feature that requires players to constantly evaluate the position of all players and switch between defensive and aggressive play.

Due to the high degree of incomplete information in the beginning of the game and the number of players, the state space describing possible evolutions of the game is vast. This essentially renders the traditional search techniques such as AB trees or Minimax applicable only in the final stages of the game where a lot of cards have been discarded. On the other hand, reactive strategies based on the current state and the last executed actions offer a different approach to decision making for non-player characters in other game genres than card games that has proven useful. More specifically, finite state machines (FSMs) offer a simple way to specify a reactive behavior. An FSM is defined by a finite number of states, each of which intuitively characterizes a specific type of pre-defined behavior. Changes between states occur through triggered conditions corresponding to discrete actions or events during the game.

Our work intends to combine strategic play with deliberative search techniques in order to provide useful guidelines for an AI Mini-Tichu player, for the greater part of the game that the state space is not manageable solely by search.

2. OUR APPROACH
Our approach is based on the development of different components that account for several aspects of the game combining search-based methods and strategic play. We use the term *Hand* to refer to the cards a player is holding at any instance of the game. A *Combination* is a set of cards that qualify as a valid combination according to the rules of the game and a *Decomposition* of a hand is a set of combinations such that all cards of the hand are used in exactly one combination. The components of our approach are the following:

• **Identifying all the possible card combinations for a given hand.** A tree structure is constructed from the hand

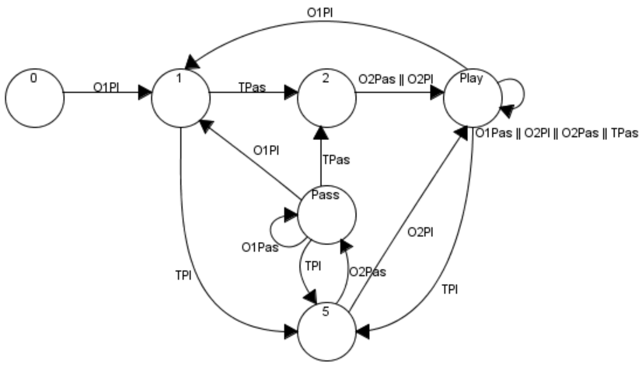

Figure 1: FSM when all four players are in the game

decomposition with all the possible combinations appearing at the terminal nodes. A five-card hand containing only a full-house combination will produce three terminal nodes, each one containing a different decomposition (two pairs and a single card, a pair and a three of a kind, and a full house).

• **Computing winning guarantees for a specific combination.** The discarded cards are being marked in a 13x4 array which corresponds to the 52-card deck.

• **Evaluating possibility of winning.** Each combination is evaluated as "trick-winning" using the array of the previous component. For example, when three Aces have been discarded, a pair of Kings is a trick-winning combination. The hand is rated as a winning one or not, according to the number of trick-winning combinations.

• **Evaluating hand decompositions.** Each terminal node is evaluated, in order to select the superior one, using as parameters the combinations' power, the number of cards of each combination and an empirical coefficient corresponding to each combination type. For example, a node containing no single cards of low value will be evaluated as stronger than a node containing such single cards.

• **Using tactical play techniques.** This component specifies some tactical play for the player based on the last actions performed by the other players. A different FSM is used at various phases of the game, e.g., when three players remain in the game. States that correspond to the AI player specify information considering his strategy using the other components. The remaining states of each diagram are considered as memory trackers for the actions of the other players.

Figure 1 represents the basic FSM diagram used when all four players are in the game. Each transition denotes the action of some player. States *Play* and *Pass* prescribe certain guidelines which must be adopted by the AI player. *T*, *O1* and *O2* indicate the AI player's partner, the first and the second opponent respectively. Transitions *Pl* and *Pas* represent *Play* and *Pass* actions. When one player runs out of cards the state diagram changes. For example, this diagram indicates a *Pass* action to the AI player in order to help his teammate when he has played and the opponent *O2* has passed (transitions *TPl* and *O2Pas*). In a different case where the opponent has played over the teammate, a *Play* behavior would be suggested (transitions *TPl* and *O2Pl*) .

Each of the components above contributes to the construction of the AI player's final behavior. The decomposition of the hand is used to define the possible combinations that can be formed with a specific hand. After having constructed the search tree each terminal node is evaluated. The node holding the maximum value is selected. After each turn a new search tree is constructed and the best node is reselected.

During the game, the strategy component is used. In each turn, according to his position in the state diagram, the AI player is able to play defensively, pass or play aggressively. During a defensive play the AI player will choose to play a combination only if it preserves the average value of his hand. For example, when a pair of threes has been played by the previous player, a pair of fours is more encouraged to be played than a pair of Aces. If there is no way to do so, the AI player would pass. In the case where an aggressive play is requested, the combination that is played is the one that is more probable to winning the trick.

This approach contributes to the construction of a challenging and multidimensional AI player who is able to take into consideration a variety of the game features.

3. DISCUSSION AND FUTURE WORK

A typical approach for solving card game problems is relying on specialized domain-dependent or sophisticated general-purpose search methods. More recent studies have used probabilistic methods such as the Monte-Carlo sampling [1] as well as Hierarchical Task Network planning [5] and equilibrium - finding methods [2] for popular card games such as Bridge and Scat. Minimax search with AB pruning is probably the most widely used technique, for which optimality in certain games has also been proven [3, 4]. The intention is that the proposed approach works in collaboration with such search methods, providing "approximate" guidance until it becomes possible to use such search methods.

Our current evaluation using a Java-based implementation shows very promising results. In particular, for controlled experiments we are able to verify that the reactive part actually makes the difference guiding the player to pass or attack (in turn using the deliberative search method) depending on the state of the game and the response of their team-mate and opponents. A web-based platform is being developed on top of this implementation in order to attract a large number of players and test the different AI player configurations in a more challenging setting.

4. REFERENCES

[1] I. Frank and D. Basin. A theoretical and empirical investigation of search in imperfect information games. *Theoretical Computer Science*, 252:217–256, 1999.

[2] A. Gilpin. Finding equilibria in large sequential games of imperfect information. In *In ACM Conference on Electronic Commerce*, pages 160–169, 2006.

[3] J. Pearl. The solution for the branching factor of the alpha-beta pruning algorithm and its optimality.

[4] A. Plaat, J. Schaeffer, W. Pijls, and A. D. Bruin. Exploiting graph properties of game trees, 1996.

[5] S. J. J. Smith, D. Nau, and T. Throop. Computer bridge: A big win for ai planning, 1998.