6th International Conference on Applied Informatics
Eger, Hungary, January 27–31, 2004.

Games with few players

Gergely Lakatos[a], Benedek Nagy[b]

[a] University of Debrecen, Debrecen, Hungary
e-mail: gergo_gent@freemail.hu

[b] Department of Computer Science, University of Debrecen, Hungary,
Research Group on Mathematical Linguistics,
Rovira i Virgili University, Tarragona, Spain
e-mail: nbenedek@inf.unideb.hu

Abstract

In this paper we investigate the multiplayer games with few participants. In these games we have less participants as the usual in n-player games. For this reason we use a variation of the methods which are used for the 2-player zero-sum strategic games. Some definitions are given analogously and in some cases more definitions are needed because loosing the special case of 2 persons. The game-trees are built with the so-called expected outcome-vectors. The situation – comparing to the 2-player games – looks more like the non-zero-sum games, because more values are needed to calculate the "optimal" strategy.

Assuming at least 3 players in a game it is not guarantied that any of the players has a non-loosing strategy. In this paper we present some methods (based on the well-known minimax algorithm) for evaluation of the games. Methods to help for making decisions for the players are also shown both with partial and full evaluation of the game-tree. The question of "rational strategy" is interesting in these games, as for example at the prisoner's dilemma.

Categories and Subject Descriptors: I.2.1 [Artificial Intelligence]: Applications and Expert Systems - games; G.2.2 [Discrete Mathematics]: Graph Theory - trees

Key Words and Phrases: game theory, artificial intelligence, multiplayer game trees

1. Introduction

The game theory and the artificial intelligence approach the problem of 2-player zero sum games in a different way. The game theory ([6,8]) uses pay-off matrices to determine the pay-offs of the players at the end of the games (matrix games). In case of complicated games this matrix can be too large to construct. The artificial intelligence uses the graph approach ([1,7]): the main aim is to construct game trees and use some evaluation methods to find out the best strategy. When we try to represent a complicated game it is not possible to construct the whole game tree. In these cases the partial-evaluation is used, combined with heuristic techniques.

2. The class of games with few players

In this paper the word **game** means finite and constant(usually zero)-sum strategic game with at least 3 players. For every game there is an integer k ? 3 which is called the **number of players**. Note that for all non constant-sum games can be extend to zero sum games adding a new player who has no real choice, but get a pay-off to let the game zero(fix)-sum.

Definition 2.1. We call a game **fixed player order game** if the players are only allowed to move one after the other in a fixed order.

Remark 2.1. For a fixed player order game it stands that every player has a possibility to move every k-th step.

The **set of pay-offs** an ordered set of vectors which contains all the possible points that the players can receive at the end of the game in a respective order. We call the i-th player's **pay-off** the decimals which are the i-th elements of the vectors of the set of pay-offs.

Definition 2.2. Let's indicate $H = \{h_1, h_2, \ldots, h_n\}$ the **set of p** ossible game states which is a set that contains all the game states that can occur during any game according to the rules of the game. $P = \{p_1, p_2, \ldots, p_k\}$ indicates **set of players**.

In a fixed player order game usually the indices i $(1 \leq i \leq k)$ mean the order of steps during the game.

Definition 2.3. Thet **state-space** is the set defined as follows: $A = \{(h, p)|h \in H,\ p \in P\}$, where h denotes the actual game state and p is the player to move next. The **set of initial states** stands of pairs from $H \times P :\ K = \{(h, p_1)|h \in H,\ start\_cond(h),\ p \in P\}$, where h satisfies the starting condition and p_1 is the starter player. Note that $K \subseteq A$. The set of **final states** also contains pairs from $H \times P :\ V = \{(h, p_i)|h \in H,\ end\_cond(h)\}$, where h satisfies the end condition and p_i denotes the player that would have moved if the game hadn't ended.

Note that in some games according to the rules p_i is the loser of the game.

Definition 2.4. We call the **set of steps** the following set that contains the possible moves:
$L = \{l : h_1 \to h_2 | h_1, h_2 \in H,\ precond\_l(h_1)\}$, where l is a mapping from H to H satisfying it's own precondition of use. The **set of operators** stands of functions which are defined to make a mapping from $H \times P$ to $H \times P:\ O = \{o : (h, p_i) \to (h', p_j) | h, h' \in H,\ l \in L, precond\_l(h), l : h \to h'\}$. At fixed order player games $j = i + l$.

Definition 2.5. In t his way we obtain the **state-space representation of the game** as follows: $J =< A, K, V, O >$, where A,K,V,O are the previously defined corresponding sets.

3. Winning strategy, non-loosing strategy

Definition 3.1. Strategy is a function of one particular player wich determines an appropriate step in every game state.
$STRp_i : h \to l$, where $h \in H$, $l \in L$ and $(h, p_i) \in A$.

As we have already mentioned every player receives some points at the end of the game. Using the ordered property of pay-off vectors we can define the ranking of the players at the end of the game.

Definition 3.2. Let m indicate the number of different pay-offs at the end of the actual game ($1 \leq m \leq k$). Put these pay-offs in a non-descending order. A player **ends in the i-th place** ($1 \leq i \leq k$) at the end of the game if his/her pay-off is i-th in this order. Note that there can be more than one player ending in the same place. The **winners of the game** are the players who finish in the first place. The **losers of the game** are the players who finish in the last place. The game ends with **draw** when all the players receive the same points so they all finish in the same place.

Definition 3.3. One player has a **winning strategy** in a few player game if by playing this strategy he becomes the winner of the game whatever strategies the other players choose. A player with winning strategy is called **strong dominant player**. One player has a **non losing strategy** in a few player game if by playing this strategy he receives at least as much points as the sum of all the pay-offs at the end of the game whatever strategies the other players choose. A player with non loosing strategy is called **dominant player**.

We recall from the theory of 2-player games the theorem that in all zero-sum 2-player games there is at least 1 player who has a non-loosing strategy, and when the result of the game cannot be a draw, then a player has a winning strategy ([2,3,6]). As we will show similar statement does not hold for few player games.

4. The graph representation

It is always easier to understand and solve a problem if we are able to visualize it. This leads us to the graph representation.

Definition 4.1. The following structure is called the **graph representation** of a game: $G =< N, S, T, E >$, where

- N: is the set of nodes of G,
- S: is the set of startnodes (with only outgoing edges),
- T: is the set of endnodes, leafs (with only incoming edges),
- E: is the set of edges.

The graph representation of a game is usually called **game-graph**.

Formally, if we have the state-space representation $< A, K, V, O >$ of a game than we can easily obtain the graph representation $< N, S, T, E >$ only by showing how the sets of the different representations correspond each other:

- The elements of A are visualized by the elements of N.
- The startnodes of $S \subseteq N$ belong to the final states of K.
- The elements of V are related to the elements of T.
- The edges of E represent the operators of O.

Definition 4.2. A game-graph is called **game-tree** if the following conditions are satisfied:

- $|S| = 1$, so there is exactly one startnode,
- every node has exactly one incoming edge, except the startnode.

Definition 4.3. A chain of nodes and edges starting with a startnode ending with a leaf is called a **path** in a tree. A path is always part of a tree.

A path corresponds to a possible game. In a fixed order player game possible steps of each level corresponds to a player.

Definition 4.4. A game-tree is called **whole game-tree** if all the possible paths starting with one particular startnode are part of it.

We can derive and-or tree-graphs for any general game-tree. Only the edges coming out of a node belonging to our chosen player become or-edges and all the others become and-edges.

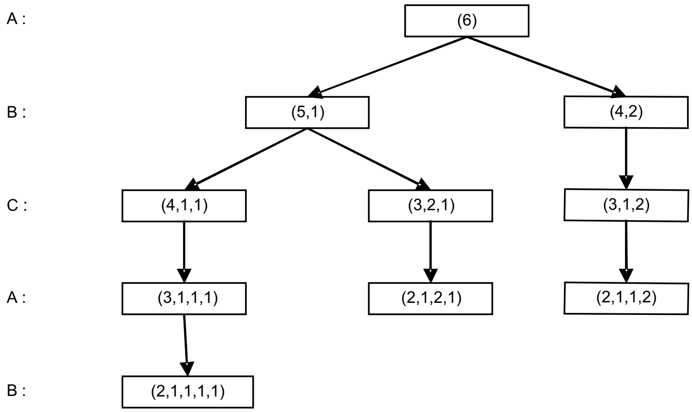

Figure 1: The whole game tree of the 3-person
Grundy-game starting with 6 coins.

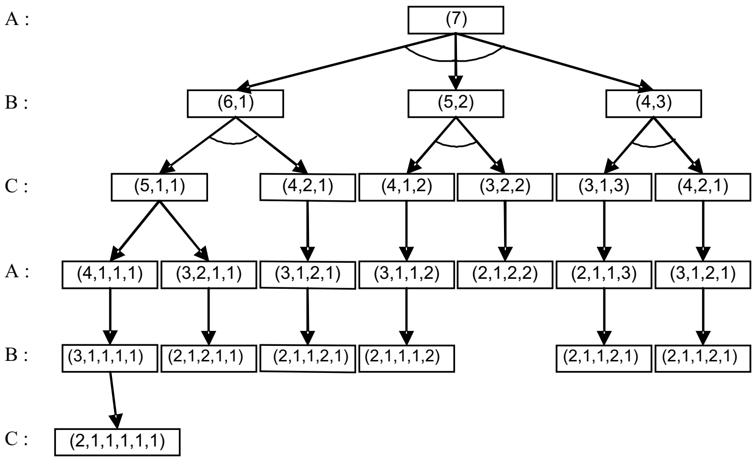

Figure 2: The and-or tree of the 3-person
Grundy-game for “C”, starting with 7 coins.

5. Evaluation techniques

In this section we present some evaluation method from a chosen player's viewpoint. (This approach looks like the approach of game-theory, where a chosen player is important; all others are used as a statistical background.) The evaluation techniques are realised by using different labelling methods. The main point is to start from the leaves: the pay-off of the actual leaf node belongs to the node as a label. Then the parents get their label by applying some kind of function on their children nodes' labels. By continuing this method, at the end the startnode also gets a label. Here are the different labelling methods.

5.1. Optimist evaluation

It is based on the well-known graph-search methods (breadth-first, deep-first [1,7]). The main point is to choose the best possible pay-off for any node in the tree. It can be used for any-player games, including 1-player ones. (see Fig. 3.a)

Example 5.1. Two parents play with a child. All of them want the child's win.

5.2. Pessimist evaluation

Opposite to the previous method, the chosen player is alone, and all the others want to beat him/her. Formally if the chosen player has the possibility to move, choose the best of the children nodes' labels. But in case of any other player's turn choose the worst. This method gives the guaranteed minimal pay-off for the chosen player. (see Fig. 3.b) The evaluation can be partial using a variation of the 2-players alpha-beta cut algorithm based on pessimist evaluation: alpha for the chosen player, beta for the others. It is efficient in games where the players have 2 groups, for instance ultimo, tarok (see [4]), etc.

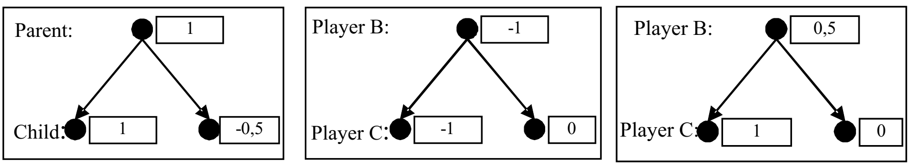

Figure 3: a. Optimist evaluation, b. Pessimist evaluation for player "A", c. Average evaluation

5.3. Average evaluation

Formally if the chosen player has the possibility to move, choose the best of the children nodes' labels. But in case of any other player's turn count the average of the labels. Easy to evaluate it, it is more realistic than the previous two methods. (see Fig. 3.c)

5.4. Depending on others evaluation

It is a mixed technique using the previous three evaluations. Our aim is to make groups of players. First group is the chosen player and the friends. All the enemies belong to the second group. The third group contains the others who are neutral. If someone from the first group has the possibility to move, use the optimist evaluation. If it is the turn of someone from the second group use the pessimist. In case of a step of a player from the third group use the average evaluation.

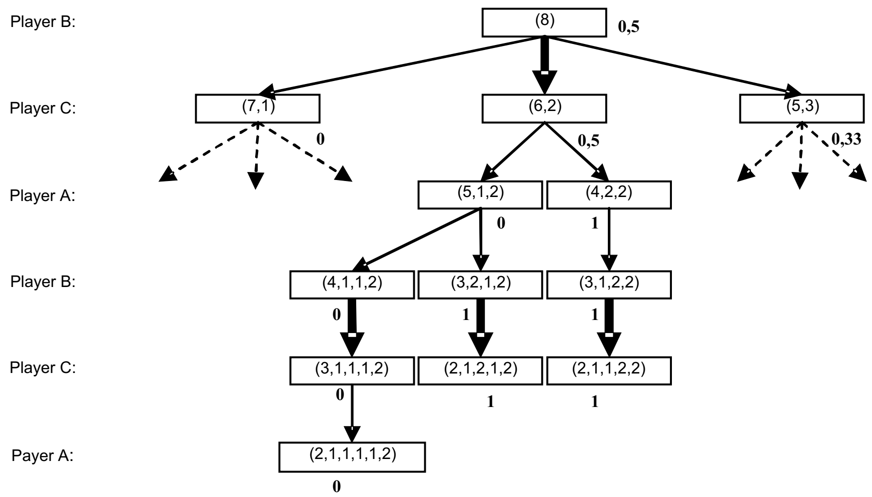

Figure 4: The depending on others evaluation.
Chosen player: “B”, enemy: “A”, neutral: “C”.

6. Labelling by vectors

The basic idea is to use all the player’s pay-offs by creating pay-off vectors. So, there is not a chosen person, but we want to have the evaluation for all players. This way we can use more information so we get more sophisticated and more realistic results but on the other hand it takes more time to evaluate the game tree.

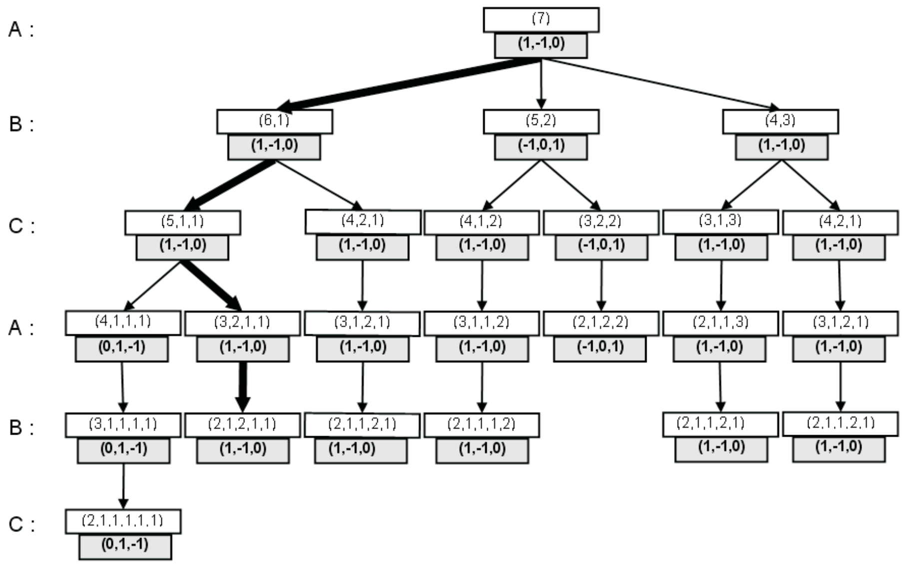

Figure 5: The best-choices evaluation.

6.1. Best-choices evaluation

When choosing the label of a node from the children's nodes we only watch the pay-off of the player who is about to move: we choose the label which contains the most points for him/her. This method can be used by labelling the final place of the game instead of the points as well. In this way all players can predict the game, but this 'best-choice' game may not be the best for all of them. It is possible that this actual game is not 'rational' for all the players (related to [5]).

6.2. Guaranteed result evaluations

Using the pessimist evaluation parallel for each player using vectors, we get the guaranteed pay-off vector at the startnode. Using the end-places for this evaluation we prove the following theorem, which means that the situation differs from the 2-player case.

Theorem 6.1. *In a constant-sum few player game not even the existence of a dominant player is certain.*

Proof. *By using the evaluation method above for all three players, the example given in Fig. 6 presents a game where is no dominant player. At the startnode there are negative points in each vector for the corresponding player.*

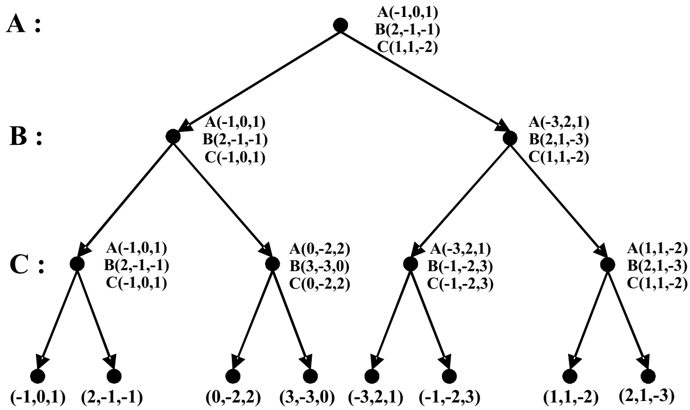

Figure 6: The guarantied best end position evaluation for all three players in the same time.

For 2-player games there is no different in evaluations by points and evaluation by ending-places. For more player games a different can occur. For example what does player A prefer if he can choose between payoff vectors (2,1,-3) and (3,-7,4). The first one is better if he wants to win, but the second one means more points.

7. Mixed techniques

7.1. Evaluation by more aspects

When choosing the best (or worst) label of one node's children nodes it is worth using different aspects. First make an order of the aspects starting with the most important one followed by the second one and so on. This is a kind of function of aspects where we use only as many aspects as necessary to be able to get precisely one label which is the best (or worst). One can also use other kind of functions, for instance weighted averages of the aspects to get a more sophisticated evaluation.

7.2. Long-term evaluation (repeated games)

Sometimes the long-term interest of a player is different from the short term interest.

Example 7.1. Better if I am the 2nd and P is the fifth than I am the winner and he is the second because he is my long term enemy as his summarized points of the previous games is the closest.

That's why in some cases we don't have to watch some player's pay-offs if their summarized points are very low (they won't catch up with us) or too high (impossible to reach them) compared to ours. On the other hand it is worth trying to focus on such players whose summarized points are very close. The situation is the following: one wants to play against somebody and not only for (enjoy) the game itself.

8. Conclusions

As we showed the theory of few-player games is more complex. We showed evaluation methods using only a chosen player's point of view and methods including all players' aims. We tried to develop some techniques for analysis of few-player games based on the well-known graph/tree approach for 2-player games. The theory can be extended in more way depending on the aims of the players: getting more points or resulting better places in the end of the game can be cause different results. The situation looks like the non-zero sum 2-player games. Moreover, in many cases the psychology plays an important role among the players.

References

[1] I. Fekete, T. Gregorics and S. Nagy: *Bevezetés a mesterséges intelligenciába*, *Bevezetés a mesterséges intelligenciába. - Heurisztikus gráfkeresés, kétszemélyes játékok, automatikus tételbizonyítás*, LSI Oktatóközpont, Budapest, 1990.

[2] L. Filep: *Játékelmélet*, Filum Kiadó, Tankönyvkiadó, 1985.

[3] R. D. Luce and H. Raiffa:*Games and decisions*, Dower, 1989.

[4] M. Lustrek and M. Gams: *"Intelligent system for playing tarok"*, Journal of Comp. And Inf. Techn. – CIT, **11**, (2003) 209–215.

[5] L. Mérő: *Mindenki másképp egyforma. A játékelmélet és a racionalitás pszichológiája*, Tercium 1996.

[6] M. J. Osborne and A. Rubinstein: *A course in game theory*, MIT, 1994.

[7] S. J. Russel and P. Norvig: *Artificial Intelligence. A Modern Approach.* Panem/Prentice Hall, 2000.

[8] J. Szép, F. Forgó and F. Szidarovszky: *Introduction to the Theory of Games: Concepts, Methods, Applications*, Kluwer, 1999.

Postal addresses

Gergely Lakatos
University of Debrecen,
4032 Debrecen, Poroszlay út 24. 3/3.
Hungary

Benedek Nagy
Department of Computer Science,
University of Debrecen,
Debrecen, PO box 12.,
Hungary