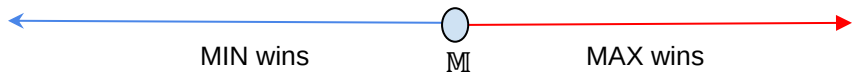
Activity Based Gaming: Multiplayer Games

Sarang Aravamuthan (sarang@alumni.iitm.ac.in)

In an earlier article [1], I had described a procedure for converting any activity into a two-player game provided that

1. The activity can be completed in a finite number of steps and
2. With each possible completion, we can associate a (numeric) value.

The players MAX and MIN take turns to complete the activity but with contrasting goals of maximizing and minimizing the final value. Then, under *optimal play* from both players, the final value evaluates to what we call the *minimax value* of the game (hereafter denoted by $\mathbb{M}$). Under normal play, the final value is compared with $\mathbb{M}$ to decide the winner.



The figure above illustrates the different scenarios under which MAX or MIN can win. In particular, MAX (MIN) wins when the final value is larger (resp. smaller) than $\mathbb{M}$.

As an example of an activity, consider evaluating the arithmetic expression

$$f(X,Y,Z) = X(Y-Z).$$

MAX and MIN want to maximize (resp. minimize f). Towards this end MAX proposes digits (0—9) that MIN substitutes for a (un-instantiated) variable of her choice. The activity is completed when all variables are instantiated and the value on completion is the value of f. For example, one possible sequence of moves is

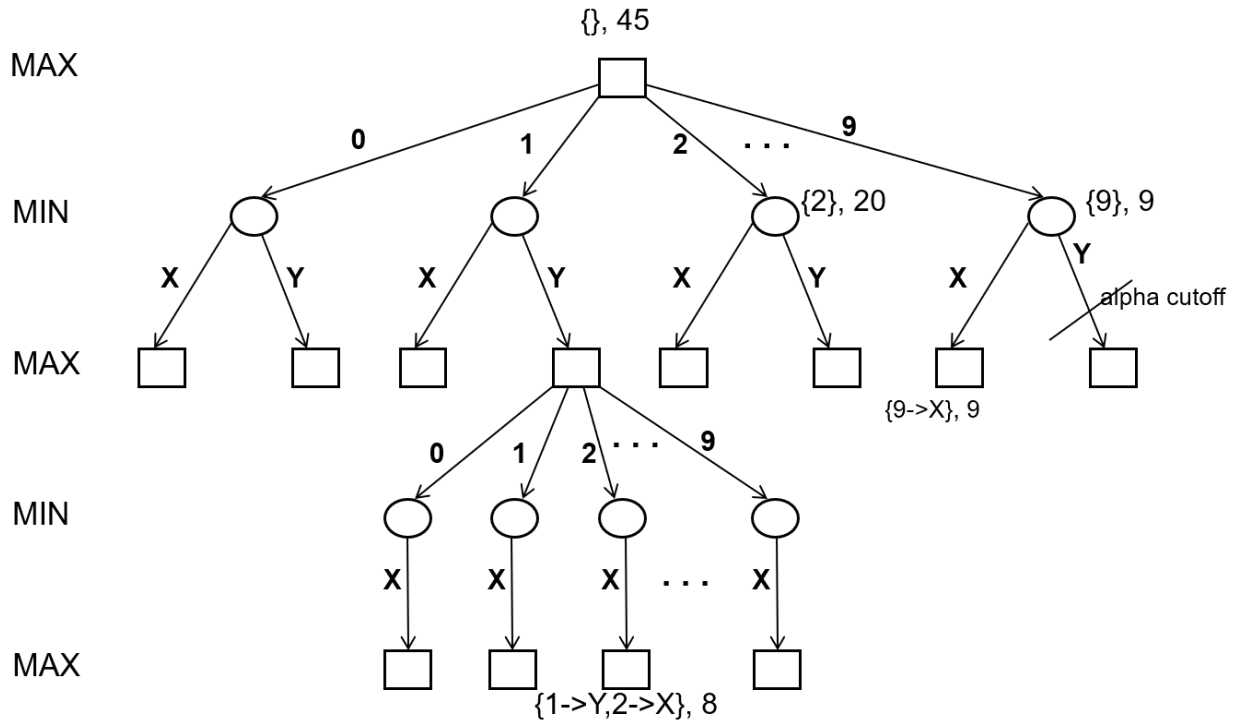
1. MAX proposes 4 which MIN substitutes for Y
2. MAX proposes 2 which MIN substitutes for Z
3. MAX proposes 3 which MIN substitutes for X

leading to the final value $f(3,4,2) = 6$.

What's $\mathbb{M}$ here? If MAX starts by proposing a high number then MIN would substitute that for Z making the expression small. On the other hand, a low choice would be substituted for X. After some trial and error, the reader can convince herself that $\mathbb{M} = 18$. In the example above, MIN wins since 6 < 18.

Let's pause to consider what we mean by optimal play and minimax value. Informally $\mathbb{M}$ is the best outcome that MAX (and MIN) can hope for i.e. for any $v > \mathbb{M}$, MIN has a strategy that ensures that the final value $< v$. Similarly for any $v < \mathbb{M}$, MAX has a strategy that ensures that the final outcome $> v$.

$\mathbb{M}$ is evaluated by tracing the states of the game through a game tree (a sample tree is shown below) and using the minimax algorithm. This is computationally intractable even for games of moderate complexity; see [1] and [2] for more details and further examples of two-player games.



A (partial) game tree for the expression $f(X, Y) = (10 - X)Y$. Each node is labeled with the sequence of moves leading to that position and the minimax value under optimal play from thereon.

Here are a few other aspects of two-player games.

1. An analogy: We can interpret a two-player game as the mental analogue of the classical game of Tug-of-war with $\mathbb{M}$ playing the role of the *center line*. The players pull the activity value towards their side with all their computational might and the winner is determined by which side of $\mathbb{M}$ the final value falls in.

2. First play: A noteworthy aspect of using this rule to define the winner is that it negates the first player advantage provided by some games such as chess. If the final value at the completion of game is above average (i.e. skewed towards the max), then this is encoded in a higher minimax value and MAX has to score higher than this to win.

3. A minimax interpretation of chess: Interestingly, the game of chess can also be placed in a minimax setting. Here the activity is the removal of a king (of either color) and the

value at completion of the activity is the parity of the number of moves, i.e. 1 if the black king is removed and 0 otherwise. Thus MAX and MIN, who move white and black pieces respectively, play to ensure that the game ends on their move. To consider drawn situations (such as only kings left on board), we need to put a limit on the number of moves and if the activity exceeds this limit then we terminate it with a value of 0.5. But the big question here is "what's $\mathbb{M}$"? Conventional wisdom suggests that $\mathbb{M} = 1$ since white is believed to have a slight advantage. However if this is proved (which seems improbable considering the complexity of the game), then under the minimax version, MAX would never win since $\mathbb{M}$ is the maximum possible value!

On to Multiplayer Games

Unfortunately a scalar, like a rope, can be only pulled along two directions, $+\infty$ and $-\infty$. However a point even in $\mathbb{R}^2$ can be pulled in infinitely many directions. The setting we define is as follows:
Say we have N players $p_1, \cdots, p_N$ $(N > 1)$ and an activity A that can be completed by a sequence of moves. The activity can be completed in different ways and with each possible completion, we associate an m-dimensional value $(m > 1)$ that we'll call $Y \in \mathbb{R}^m$. The players take turns in making the moves. With each p_i, we also associate a *direction vector* $v_i \in \mathbb{R}^m$ and p_i's goal in making the moves is to maximize the component of Y along v_i. In other words, p_i wants to maximize the *dot product* $Y \cdot v_i$. Under optimal play from each player, Y evaluates to the minimax value $\mathbb{M}$ (we show below how this is computed). Under normal play, the difference between $Y$and $\mathbb{M}$ is computed and its component evaluated along each v_i. The *payoff* for $p_i$is $(Y - \mathbb{M}) \cdot v_i$ and the winner is the player with the largest payoff. Further, the players can be ranked by their payoffs.

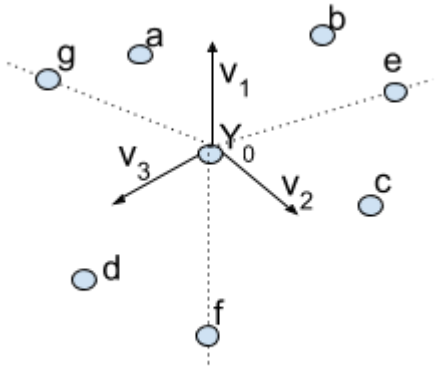


Figure 1: A 3-player game

The figure above illustrates this with 3 players (and associated direction vectors $v_1, v_2$and v_3) with final values being evaluated on the plane ($N = 3$, $m = 2$). One of these is the minimax value (indicated by Y_0 in the figure). The dashed lines demarcate the plane into regions of victory for each player. Thus if the final value is

- a or b ⇒ p_1 is the winner
- c ⇒ p_2 is the winner
- d ⇒ p_3 is the winner
- e ⇒ $p_1$and p_2 are joint winners
- f ⇒ $p_2$and p_3 are joint winners
- g ⇒ p_1 and p_3 are joint winners
- Y_0 ⇒ its a 3-way tie.

Computation of minimax value

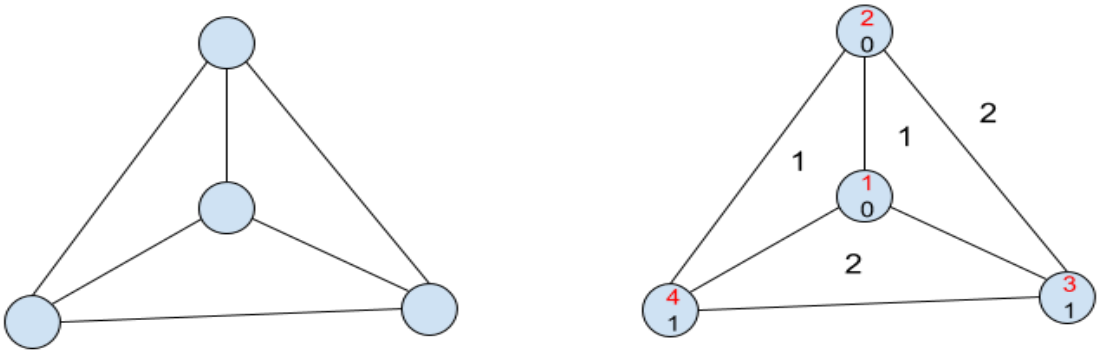
We assume that the players make their moves in order $p_1, \cdots, p_N, p_1, \cdots$ with $p_1$making the first move. As in the two-player version, we use a *game tree* to represent the moves of the players (see [3] for an illustration). The nodes of the tree correspond to the different stages of the partially completed activity. The root node represents the start of the activity and the edges denote the possible moves a player can make. Each node is labelled with the player whose turn it is to play from that position. We also label each node with a (m-dimensional) value. This is the minimax value associated with the position corresponding to that node. In other words, this is the value the activity would evaluate to under optimal play starting from that position. This value is computed in the following way.

The terminal node is assigned the value of the completed activity. These values are then backed up the tree level by level. Given an internal node labelled p_i, suppose its k children have been assigned values $y_1, \cdots, y_k$. p_i will make the move that maximizes the component of the final value along v_i. In particular, if $y_j \cdot v_i = max(y_1 \cdot v_i, \cdots, y_k \cdot v_i)$, then the value at this node is y_j. As in the two-player version, the value at the root node is $\mathbb{M}$.

Sounds simple right! Well, there's small glitch here. Namely the uniqueness of the values that bubble up the tree. While the maximum of a set of numbers is unique, the same cannot be said when we are computing the maximum w.r.t. a vector. In other words, there may be several values whose dot product with v_i is the same. Then the question arises, which value is backed up? The only resolution I see here is to choose the direction vectors v_i in such a way that the scalars $v_i \cdot y$ are distinct over all possible values y of the completed activity.

Example 1: Here's a simple 3-player game illustrating this scheme.
In the figure shown below, players p_1, p_2, p_3 take turns entering values (0, 1 or 2) in the circles. The activity is completed when all 4 circles are filled. The triangular faces and the outer face are then filled with numbers 0, 1 or 2. The number in a face is the sum of the numbers in its vertices mod 3. The final value is the point $(n_0, n_1, n_2) \in \mathbb{R}^3$ where n_i is the number of faces labeled $i, i = 0, 1, 2$. The goals of p_1, p_2 and p_3 are to maximize n_0, n_1 and n_2 respectively (so their

direction vectors are $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$).



An instance of such a play is shown in the figure on right. The players fill the circles numbered 1 to 4 (in that order) with values shown. The final value as we can see is (0, 2, 2).

To find $\mathbb{M}$, we may assume by symmetry that the inner circle is filled last. If the 3 outer circles sum pairwise to distinct values mod 3, then the outer face will be labeled 0 while any choice for the last entry will lead to distinct labelings for the 3 inner triangles. This leads to a final value of (2, 1, 1). If the pairwise sums are all the same, then p_1 can ensure that all 4 faces are labelled 0 but this is clearly due to suboptimal play from p_3 since p_3 can ensure that at least one triangle is labeled 2. Otherwise the pairwise sums are of the form (x, x, y) where $x = 0, 1$ or 2 and $y \neq x$. In such a case $p_1$chooses $3 - x$ for the last entry so that the final (minimax) value is (2, 1, 1).

For the play above, the difference $= (0,2,2) - (2,1,1) = (-2,1,1)$ and the payoffs for the players are -2, 1 and 1. Thus in this case $p_2$and $p_3$are joint winners.

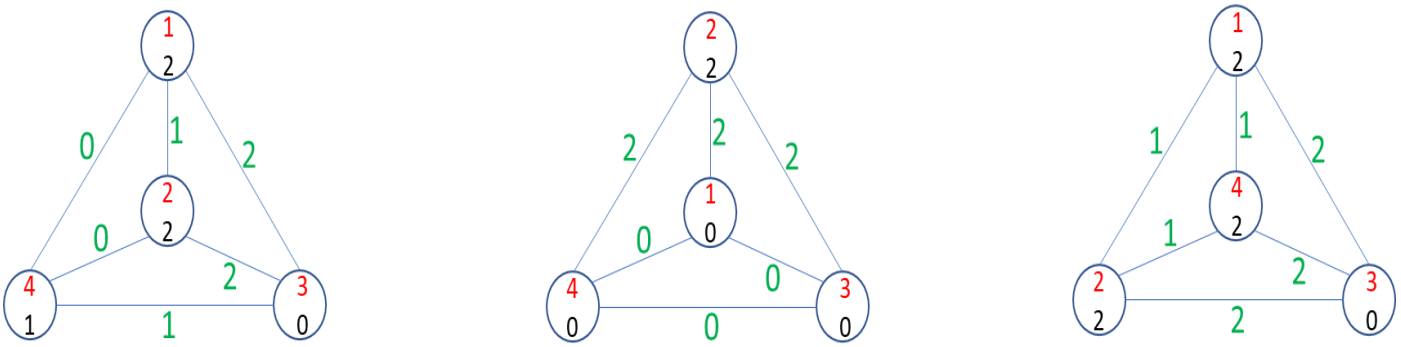
Avoiding the Computation of $\mathbb{M}$

We have seen that finding $\mathbb{M}$ is computationally intractable even for games of moderate complexity. Surprisingly it's possible to rank the players without explicitly calculating $\mathbb{M}$. The idea is to complete the activity N times (instead of just once). Each player pulls along a different direction vector in each run.

To make matters precise, say we have N players and direction vectors $v_1, \ldots, v_N$. Let $v = \sum v_i$. Suppose the activity values on N completions are $\mathbb{x}_1, \ldots, \mathbb{x}_N$. If a player p pulls along directions $v_1, \ldots, v_N$ in sequence, then net payoff for $p = \sum_{i=1}^{N} (\mathbb{x}_i - \mathbb{M}) \cdot v_i = \sum_{i=1}^{N} x_i \cdot v_i - \mathbb{M} \cdot v$.
The last term is common for all players and can be canceled out.

Let's illustrate this with an example.

Example 2: Coloring a graph thrice



Players p_1, p_2, p_3 label the vertices of a graph 0, 1 or 2. The order in which the vertices are labeled is shown in red and the actual labels are in black. After all vertices are labeled, the edges are colored 0, 1 or 2 where the color of an edge (u, v) = (label(u) + label(v)) mod 3.

The value at the end of the activity is the point $(n_0, n_1, n_2) \in \mathbb{R}^3$ where n_i is the number of edges colored $i, i = 0, 1, 2$. The direction vectors are $e_1 = (1, 0, 0), e_2 = (0, 1, 0)$ and $e_3 = (0, 0, 1)$.

In each run the first player pulls along e_1, the 2nd along e_2 and the 3rd along e_3 (i.e. the first player wants to maximize the number of 0-edges…).
In the 3 runs, the players play in sequence $(p_1, p_2, p_3), (p_3, p_1, p_2), (p_2, p_3, p_1)$.

From the figure above, we see that the final values in the 3 runs are $(2, 2, 2), (3, 0, 3)$ and $(0, 3, 3)$.
Then net payoff for $p_1 = 2 + 0 + 3 - \mathbb{M} \cdot e = 5 - \mathbb{M} \cdot e$ where $e = e_1 + e_2 + e_3 = (1, 1, 1)$.
Similarly, net payoff for $p_2 = 2 + 3 + 0 - \mathbb{M} \cdot e = 5 - \mathbb{M} \cdot e$ and
Net payoff for $p_3 = 2 + 3 + 3 - \mathbb{M} \cdot e = 8 - \mathbb{M} \cdot e \Rightarrow p_3$ wins.
and we didn't have to compute $\mathbb{M}$ for that!

Comparison to the Two-player Game

We close this article with some practical issues that arise when designing multiplayer games.

Choice of direction vectors: The multiplayer game is defined by the activity, the moves that complete it and the function mapping each completion to a value. However the direction vectors define the *incentives* for the players in completing the activity. In the two-player game, the directions are usually along the positive and negative X-axis and $\mathbb{M}$ is a function of the rules for making the moves. In the multiplayer version, $\mathbb{M}$ depends additionally on the direction vectors.

In Figure 1, we showed the region of victory for each player in a 3-player game. Thus ideally the directions should be chosen in such a way that there are roughly the same number of values in each region. In practice enumerating all final values is infeasible. So one solution is to simulate some game completions and use the resulting values to define the direction vectors.

There are also practical considerations in choosing the direction vectors. A goal of maximizing a component of the final value is more comprehensible to a player than maximizing along a direction that is a function of the components. For this reason alone, it's usually a good idea to choose the direction vectors to be along the coordinate axes.

Approximation of $\mathbb{M}$: We have already seen the complexity of computing $\mathbb{M}$ in the two-player case. The multiplayer version is no different and fast heuristics are desirable for good approximations to this value. For instance, a version of alpha-beta pruning for multiplayer games would allow us to evaluate $\mathbb{M}$ for games of greater depth.

Collusions: In the two-player version, one woman's gain is another man's loss. What we mean by this is that the choice of directions ensures that any move that's good for a player is necessarily bad for the opponent. In contrast, in the multiplayer version, two players can collude against a third party to curtail his chances of winning. Detecting such collusions is another challenge faced in a multiplayer game.

Multiple optimal play: In the two-player version, optimal play from one and sub-optimal play from the other will ensure a win for the former. However in the multiplayer game, the question of winner arises if two players play optimally but the others play sub-optimally.

Cooperative games: Games can be cooperative when the direction vector is same for all players. In this case, the players cooperate in completing the activity with the goal of maximizing the final value along the common direction. In such a situation, the question arises as to how the contribution to the final value is apportioned among the players.

References

[1] Activity based gaming: LinkedIn article, https://www.linkedin.com/pulse/activity-based-gaming-sarang-aravamuthan

[2] "e-Valuate: A Two-player Game on Arithmetic Expressions", Sarang Aravamuthan and Biswajit Ganguly, CoRR abs/1202.0862: (2012).

[3] Artificial Intelligence: A Modern Approach (3rd ed.), Chapter 5, Stuart Russell and Peter Norvig, Pearson education