



<http://www.diva-portal.org>

This is the published version of a paper presented at *NICOGRAPH International 2014, Visby, Sweden, May 2014*.

Citation for the original published paper:

Fridenfalk, M. (2014)

N-Person Minimax and Alpha-Beta Pruning.

In: *NICOGRAPH International 2014* (pp. 43-52).

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-235687>

N-Person Minimax and Alpha-Beta Pruning

Mikael Fridenfolk
Department of Game Design
Uppsala University
Visby, Sweden
mikael.fridenfolk@speldesign.uu.se

Abstract—This paper presents an N-person generalization of minimax aligned with the original definition. An efficient optimization method is further presented as a result of a straightforward mathematical extension of alpha-beta pruning to N-person games.

Keywords—alpha-beta pruning; dihedral angle; hypermax; minimax; multiplayer; N-max; N-person; N-player; simplex; strategic games; zero-sum lemma

I. INTRODUCTION

Games find applications in areas such as recreation, education, exercise, simulation and conflict resolution. From one perspective, a game could be defined as a quantifiable conflict with at least two possible outcomes involving at least two parties with opposing interests. A conflict is closely associated with the zero-sum game where the gain of one party is reformulated as the loss for at least one other party. The foundation of game theory is based on the two-player (synonymous with two-person) minimax theorem from 1928 [13] and can be implemented by the following pseudocode:

```
Minimax (node, p)
  if (leaf) return x
  if (p is a max-player)
     $\alpha = -\infty$ 
    for each child,  $\alpha = \max(\alpha, \text{Minimax}(\text{child}, p^+))$ 
  else
     $\alpha = +\infty$ 
    for each child,  $\alpha = \min(\alpha, \text{Minimax}(\text{child}, p^+))$ 
  return  $\alpha$ 
```

Initial call: Minimax (start node, player)

where more generally in this paper, $\mathbf{x} \in \mathbb{R}^n$ with $n \in \mathbb{Z}^+$ denotes a point in *zero-sum* or *closed space* Δ and $\psi \in \mathbb{R}^N$ with $N = n + 1$, a point in *free* or *open space* Ψ [2]. A leaf (node) denotes *game over* or *reached depth limit* and p^+ *next player*. Minimax is defined for direct application in zero-sum space. As an example in chess, the heuristic utility function for player 1 could be defined as $x = \psi_1 - \psi_2$, where ψ_1 is the sum of the heuristic values of the pieces for player 1 and ψ_2 the corresponding sum for player 2.

Minimax is, in the capacity of an exhaustive search (or brute force) method, an expensive algorithm for search in deep trees. Given the (average) branching factor b , minimax is $O(b^d)$ for search of depth d . It is possible to optimize the search speed of minimax by alpha-beta pruning (in this paper defined as a *complete optimized search method*), without changing its

outcome:

```
AlphaBeta (node, p,  $\alpha$ ,  $\beta$ )
  if (leaf) return x
  if (p is a max-player)
    for each child
       $\alpha = \max(\alpha, \text{AlphaBeta}(\text{child}, p^+, \alpha, \beta))$ 
      if ( $\alpha \geq \beta$ ) break
    return  $\alpha$ 
  else
    for each child
       $\beta = \min(\beta, \text{AlphaBeta}(\text{child}, p^+, \alpha, \beta))$ 
      if ( $\alpha \geq \beta$ ) break
    return  $\beta$ 
```

Initial call: AlphaBeta (start node, player, $-\infty$, $+\infty$)

As an example, given two merchants, if the lower bound $\alpha = \alpha_1$ for the seller is above the upper bound $\beta = -\alpha_2$ for the buyer, or $\alpha_1 + \alpha_2 > 0$, we have a deadlock and thus the condition for alpha-beta pruning is fulfilled. In practice, to minimize the number of calculations, the equality line $\alpha_1 + \alpha_2 = 0$ is culled as well. The branching factor for alpha-beta pruning is $O(\sqrt{b})$ [5].

A suggestion to extend minimax to N players, called \max^N (denoted in this paper as N to avoid mix-up with $n = N - 1$) was presented in 1986 [7], see Fig. 1.

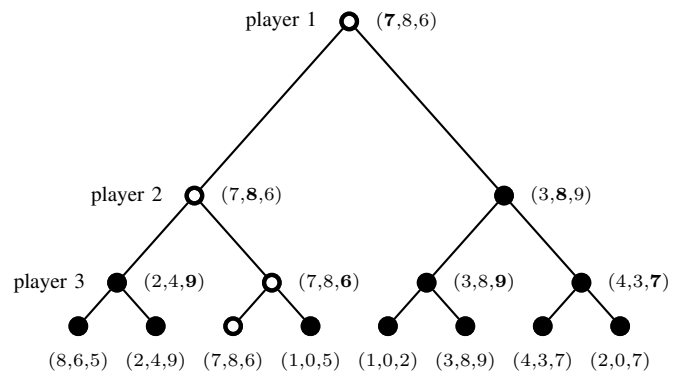


Figure 1. In \max^N , each player i of total N selects the child with the highest value associated with element i of ψ . Here with $N = 3$, $b = 2$ (children per node) and $d = 3$ (search depth).

However, since minimax is strictly zero-sum-based, \max^N does not for the two-player case in general yield the same outcome as minimax, and is thus from a mathematical point of view not an extension of minimax to N -player games,

although both algorithms have similar objectives and a similar recursive structure. The failure to find an efficient pruning algorithm for N -player games until now, may partly be attributed to this misconception.

```

MaxN(node, p)
  if (leaf) return  $\psi$ 
   $\alpha = -\infty$ 
  for each child
     $\psi = \text{MaxN}(\text{child}, p^+)$ 
    if ( $\alpha < \psi_p$ )  $\alpha = \psi_p$ ,  $\psi_{\max} = \psi$ 
  return  $\psi_{\max}$ 

```

Initial call: MaxN(start node, player)

A shallow pruning algorithm was suggested for a special case of \max^N , defined for a window of non-negative scores and an upper bound for the sum of the scores of the players. Although the branching factor was estimated as $O(b^{(N-1)/N})$ in the best case, “An average case model predicts that even under shallow pruning, the asymptotic branching factor will be b .” [6, 12]. Since the definition set of \max^N (which for each player is \mathbb{R}), is not maintained by shallow pruning, as the size of the definition window is increased, shallow pruning is steeply rendered ineffective. The same pertains to algorithms such as speculative pruning [11], that effectively fail as the size of the definition window is increased. While the branching factor of both shallow and speculative pruning are $O(b^{(N-1)/N})$ in the best case, the average complexity for speculative pruning is shown to be located somewhere between $O(b^{(N-1)/N})$ and $O(b)$, in practice closer to $O(b)$, which limits its usefulness.

In the long run, as computer technology makes further progress, even the computational complexity of games such as the perfect information game Go [3] and the imperfect information game Kriegspiel [1] will inevitably be regarded as relatively low. The reason is that even our most complex games today are only complex compared to the current speed of our computers. While the human mind has evolved rapidly from an evolutionary perspective, the pure computational speed of the human mind evolves today at a relatively slow pace compared with the evolution of computers. As an example, chess was 50 years ago regarded as a computationally complex game and it was first by the end of the last century that the best human players could be beaten by a computer (Deep Blue).

The same principle applies to games such as Go, which in the future are expected to be resolved by complete optimized search (*i.e.* primarily by alpha-beta pruning). According to one of the developers of Deep Blue, a breakthrough based on complete optimized search may for Go occur already within this decade [4]. During the last decade, the main focus of the research community has been the development of Monte Carlo methods for *incomplete optimized search* (here defined as a search method that in the general case yields a different result than exhaustive search).

Although this research may temporarily be of commercial interest, the results are however of limited interest from a long term commercial or academic perspective, since similar

methods were applied during the last 50 years in the case of chess without reaching the goal that Deep Blue finally reached by complete optimized search. It should be noted that incomplete optimized search is not exclusive for Monte Carlo methods. Quiescence search is used in combination with complete optimized search and constitutes typically the optimized high end of any complete search algorithm.

The problem with incomplete optimized search algorithms like UCT [3] is that they by definition have *blind spots*. Given access to relatively modest computational power, a Monte Carlo approach may be a good choice to start with, since the blind spots may be located further down the tree and thereby out of reach for complete optimized search, but eventually, in strategic games, the methods based on complete optimized search are always expected to catch up.

The mathematical theory in this paper is based on the n -dimensional geometric object called the regular n -simplex, see Fig. 2. A few examples are the 0-simplex (point), the 1-simplex (line segment), the 2-simplex (triangle) and the 3-simplex (tetrahedron). If the object is fully symmetric (all edges are of equal length) it is called *regular*. Scaled appropriately, the regular n -simplex exhibits the following properties:

$$\mathbf{t}_i \cdot \mathbf{t}_j = \begin{cases} 1, & i = j \\ -1/n, & i \neq j \end{cases} \quad (1)$$

$$\sum_{i=1}^N \mathbf{t}_i = \mathbf{0} \quad (2)$$

where \mathbf{t}_i and \mathbf{t}_j with $i, j \in \{1, 2, \dots, N\}$ and $N = n + 1$ denote any unit vectors i and j pointing from the center of the regular n -simplex to its i :th and j :th vertices. These properties were confirmed in [10] and [14] in context with an elementary mathematical proof of the relation $\delta = \arccos(\frac{1}{n})$, where δ denotes the dihedral angle of the regular n -simplex. For $n = 1$, $t_1 = -t_2 = 1$. For $n = 2$:

$$\mathbf{t}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{t}_2 = \begin{bmatrix} -\frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \quad \mathbf{t}_3 = \begin{bmatrix} -\frac{1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix} \quad (3)$$

In this paper, the vectors \mathbf{t}_i spanning the coordinate system of the regular n -simplex, are placed so that \mathbf{t}_1 coincides with the x -axis.

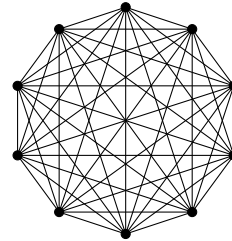


Figure 2. The n -simplex is a multidimensional object for $n > 3$ and may schematically be represented in 2D as a complete graph with $N = n + 1$ nodes. In this example, $N = 10$.

As an overview, this paper consists of a presentation of the extensions of minimax and alpha-beta pruning to N -player games, based on the derived mathematical results in Eqs. (18) and (55) and verified by systematic experiments. An important objective has in this paper been to present these derivations in the most straightforward fashion possible, but without compromising the integrity of the proofs. In the derivations of the zero-sum lemma and hypermax, proofs are first derived for the special cases of $N = 2$ and 3, and then expanded to the general case of N players. In the case of N -max, since a pedagogical proof for minimax already is presented in [9], only the general proof for N players, has been included in this paper.

II. ZERO-SUM GAMES

In the design of utility functions for computer games, **any game may be turned into a zero-sum game, if the gain of one player can be regarded as the loss for its opponent(s).**

In a game of chess for example, it is not sufficient for a player to only take its own score (the sum of the values of its own pieces) into account to make an informed decision, but the score of the opponent has also to be accounted for. For more than two players, say four in four-handed chess, it is reasonable to assume that the loss of one piece for a player in a strictly competitive zero-sum game (*i.e.* with no coalitions involved) is equal to the gain of one-third of that same piece for each of the three other players, no matter who made the capture.

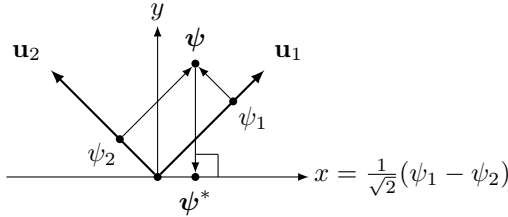


Figure 3. A two-player game where ψ^* is the orthogonal projection of a point ψ from free space on the zero-sum line $\psi_1 + \psi_2 = 0$.

In this paper, free space is defined as an orthonormal utility function-space Ψ where the independent utility functions for N players may be expressed as a point ψ in \mathbb{R}^N . Such function only considers the individual score of a player with no regard for the scores of the opponents. By contrast, zero-sum space is here defined as a subspace \mathbb{R}^n with $n = N - 1$, that in the general case constitutes the projection of ψ on the zero-sum hyperplane of Ψ such that $\sum_{i=1}^N \psi_i = 0$. Fig. 3 depicts a two-player game where ψ_i denotes the free space utility function for player i and \mathbf{u}_1 and \mathbf{u}_2 span an orthonormal coordinate system \mathbf{U}_2 symmetrically placed around y :

$$\mathbf{U}_2 = [\mathbf{u}_1 \quad \mathbf{u}_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad (4)$$

The y -axis is here only an auxiliary axis perpendicular to zero-sum space and not part of it. The orthogonal projection

of ψ on zero-sum space may be expressed as:

$$x = \frac{1}{\sqrt{2}} \mathbf{T}_1 \psi = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \frac{1}{\sqrt{2}} (\psi_1 - \psi_2) \quad (5)$$

The regular 1-simplex matrix \mathbf{T}_1 is equal to the first row of \mathbf{U}_2 multiplied by $\sqrt{2}$. In Eq. (5), x is scaled by a factor $\frac{1}{\sqrt{2}}$ which may look unfamiliar considering that the zero-sum utility function for the first player, denoted as x , is usually defined as $\psi_1 - \psi_2$ (and for the second player as $-x$). In the common case, multiplication of x with any constant real value $\kappa > 0$ does not change the outcome of minimax or alpha-beta pruning. To conclude:

$$\begin{aligned} \psi^* &= \mathbf{U}_2^T \cdot \mathbf{x}' = \mathbf{U}_2^T \cdot \begin{bmatrix} x \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} x = \frac{1}{2} \begin{bmatrix} \psi_1 - \psi_2 \\ \psi_2 - \psi_1 \end{bmatrix} \\ &= \begin{bmatrix} \psi_1 - \frac{1}{2}(\psi_1 + \psi_2) \\ \psi_2 - \frac{1}{2}(\psi_1 + \psi_2) \end{bmatrix} = \psi - \boldsymbol{\mu}(\psi) \end{aligned} \quad (6)$$

where the elements of $\boldsymbol{\mu}(\psi)$ are equal to the average value of the elements of vector ψ . Once an orthogonal projection has been performed from free to zero-sum space, ψ^* will remain in zero-sum space even if ψ^* is projected back on \mathbf{U}_2 . This shows that any two-player game that is intrinsically zero-sum based will remain in zero-sum space after orthogonal projection on x , as ψ was located on the x -axis to begin with. Since here, the y -component of the auxiliary vector \mathbf{x}' is equal to zero, only the first row of \mathbf{U}_2 is considered in the transformations. Thus for $N = 2$, the calculations may be solely based on the regular 1-simplex coordinate system $\mathbf{T}_1 = [1 \quad -1]$, such that:

$$\psi^* = \frac{1}{2} \mathbf{T}_1^T \mathbf{T}_1 \psi \quad (7)$$

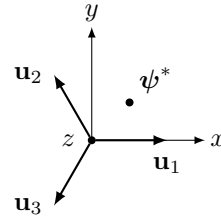


Figure 4. A three-player game where ψ^* is the orthogonal projection of a point ψ from free space on the zero-sum plane $\psi_1 + \psi_2 + \psi_3 = 0$.

Fig. 4 depicts an example where the free space utility functions of three players are projected orthogonally on the zero-sum xy -plane, with an auxiliary axis z pointing out orthogonally from the plane of the paper towards the viewer, where \mathbf{U}_3 , defined in Eq. (8), specifies an orthonormal coordinate system, symmetrically placed around z so that \mathbf{u}_1 is projected along z down on the x -axis.

$$\mathbf{U}_3 = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \mathbf{u}_3] = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \quad (8)$$

Since Fig. 4 is a 3D figure (with \mathbf{u}_1 , \mathbf{u}_2 and \mathbf{u}_3 pointing out from the plane of the paper), ψ^* gives the impression to coincide with ψ . Thus after ψ has been orthogonally projected on the xy -plane and back to free space, the following relation holds:

$$\psi^* = \frac{2}{3} \mathbf{T}_2^T \mathbf{T}_2 \psi = \begin{bmatrix} \psi_1 - \frac{1}{3}(\psi_1 + \psi_2 + \psi_3) \\ \psi_2 - \frac{1}{3}(\psi_1 + \psi_2 + \psi_3) \\ \psi_3 - \frac{1}{3}(\psi_1 + \psi_2 + \psi_3) \end{bmatrix} = \psi - \mu(\psi) \quad (9)$$

where \mathbf{T}_2 is equal to the first two rows of \mathbf{U}_3 multiplied by $\sqrt{3/2}$. The expression $\psi^* = \psi - \mu(\psi)$ is generalized below by the zero-sum lemma to any number of players $N \in \mathbb{N}_2$ (all integers equal or greater than 2).

Lemma. For any given point ψ in \mathbb{R}^N , there exists a point ψ^* in \mathbb{R}^N on an n -dimensional hyperplane with $N = n + 1$, $N \in \mathbb{N}_2$, such that:

$$\begin{cases} \psi^* = \psi - \mu(\psi) \\ \mu(\psi^*) = \mathbf{0} \end{cases} \quad (10)$$

Proof. Let us first confirm the existence of a symmetric matrix $\hat{\mathbf{H}}$ with exclusively non-zero elements that maps any point $\psi \in \mathbb{R}^N$ on a zero-sum equilibrium fix-point $\psi^* \in \mathbb{R}^N$ such that $\psi^* = \hat{\mathbf{H}}\psi^*$. The $N \times N$ matrix \mathbf{H} is defined as $\mathbf{H} = \mathbf{T}^T \mathbf{T}$, with:

$$h_{ij} = \mathbf{t}_i \cdot \mathbf{t}_j = \begin{cases} 1 & i = j \\ \gamma & i \neq j \end{cases} \quad (11)$$

where γ denotes a non-zero real value. The definition $\hat{\mathbf{T}} = \lambda \mathbf{T}$ gives the relation $\hat{\mathbf{H}} = \lambda^2 \mathbf{H}$. Assume that there exists a real value $\lambda \neq 0$, such that $\hat{\mathbf{H}} = \hat{\mathbf{H}} \cdot \hat{\mathbf{H}} = \hat{\mathbf{H}}^2$, since:

$$\hat{\mathbf{H}} = \hat{\mathbf{H}}^2 \Leftrightarrow \hat{\mathbf{H}} = \prod_{i=1}^{\Sigma} \hat{\mathbf{H}} = \hat{\mathbf{H}}^{\Sigma}, \quad \Sigma \in \mathbb{Z}^+ \quad (12)$$

The premise $\hat{\mathbf{H}} = \hat{\mathbf{H}}^2$ gives the relation:

$$\lambda^2 \begin{bmatrix} 1 & \gamma & \dots & \gamma \\ \gamma & 1 & \dots & \gamma \\ \vdots & \vdots & \ddots & \vdots \\ \gamma & \gamma & \dots & 1 \end{bmatrix} = \lambda^4 \begin{bmatrix} a & b & \dots & b \\ b & a & \dots & b \\ \vdots & \vdots & \ddots & \vdots \\ b & b & \dots & a \end{bmatrix} \quad (13)$$

where:

$$\begin{cases} a = 1 + n\gamma^2 \\ b = 2\gamma + (n-1)\gamma^2 \end{cases} \quad (14)$$

The similarity between the left and right sides of Eq. (13) gives the equation system:

$$\begin{cases} \lambda^2 = \lambda^4(1 + n\gamma^2) \\ \lambda^2\gamma = \lambda^4(2\gamma + (n-1)\gamma^2) \end{cases} \quad (15)$$

Division of both equations in (15) with λ^2 for $\lambda \neq 0$ and the second with $\gamma \neq 0$ gives:

$$n\gamma^2 + (1-n)\gamma - 1 = 0 \quad (16)$$

with the solutions $\gamma_1 = 1$ and $\gamma_2 = -\frac{1}{n}$ for $n \neq 0$. The second root is feasible and as a side effect a new elementary method has been found for the calculation of the dihedral angle of the regular n -simplex. The insertion of $\gamma = -\frac{1}{n}$ in (15) yields:

$$\lambda^2 = \frac{n}{n+1} \quad (17)$$

Thus $\hat{\mathbf{H}} = \hat{\mathbf{H}}^{\Sigma}$ for $\Sigma \in \mathbb{Z}^+$. Thereby for $\lambda > 0$:

$$\begin{aligned} \psi^* = \hat{\mathbf{H}}\psi &= \frac{n}{n+1} \begin{bmatrix} \psi_1 - \frac{1}{n}(\psi_2 + \psi_3 + \dots + \psi_N) \\ \psi_2 - \frac{1}{n}(\psi_1 + \psi_3 + \dots + \psi_N) \\ \vdots \\ \psi_N - \frac{1}{n}(\psi_1 + \psi_2 + \dots + \psi_{N-1}) \end{bmatrix} \\ &= \begin{bmatrix} \psi_1 - \frac{1}{N}(\psi_1 + \psi_2 + \dots + \psi_N) \\ \psi_2 - \frac{1}{N}(\psi_1 + \psi_2 + \dots + \psi_N) \\ \vdots \\ \psi_N - \frac{1}{N}(\psi_1 + \psi_2 + \dots + \psi_N) \end{bmatrix} = \psi - \mu(\psi) \end{aligned} \quad (18)$$

■

Below follows a proof of a previous notion on the regular n -simplex.

Proposition. The coordinate system spanning the vertices of the regular n -simplex, located at the center of the object, may be expressed as an orthogonal projection of an orthonormal coordinate system \mathbf{U} in \mathbb{R}^N with $N = n + 1$, on a hyperplane \mathbb{R}^n with $n \in \mathbb{Z}^+$, such that the column vectors of \mathbf{U} are symmetrically placed around an auxiliary vector perpendicular to this hyperplane.

Proof. Given an $N \times N$ orthonormal matrix \mathbf{U} , then $\mathbf{U}^{-1}\mathbf{U} = \mathbf{U}^T\mathbf{U} = \mathbf{I}$, where \mathbf{I} is the identity matrix. If the orthogonal projection of \mathbf{U} is performed along an auxiliary axis, perpendicular to the projection hyperplane defined by a regular n -simplex matrix $\hat{\mathbf{T}} = \lambda \mathbf{T}$, then:

$$\mathbf{U} = \lambda \left[\begin{array}{c} \mathbf{T} \\ \mathbf{c}^T \end{array} \right] \quad (19)$$

where $\hat{\mathbf{T}}$ constitutes the first n lines of \mathbf{U} and \mathbf{c} denotes a column vector of size N , where all elements are equal to a parameter c that only depends on n . The expression:

$$\lambda^2 \left[\begin{array}{c} \mathbf{T}^T \\ \mathbf{c} \end{array} \right] \left[\begin{array}{c} \mathbf{T} \\ \mathbf{c}^T \end{array} \right] = \mathbf{I} \quad (20)$$

gives the equations $\lambda^2(1 + c^2) = 1$ and $\gamma + c^2 = 0$ for $\lambda \neq 0$. Using $\gamma = -\frac{1}{n}$ and $\lambda^2 = \frac{n}{n+1}$ with $n \in \mathbb{Z}^+$, both equations give the same solutions $c = \pm \frac{1}{\sqrt{n}}$.

■

III. N-MAX

The N-max algorithm [2], applies to both perfect and imperfect information (including chance based) games. In the latter, pure-strategy is in practice extended to mixed-strategy by the replacement of score with expected value (probability multiplied by score). With the exception of the zero-sum

lemma, similar theorems have been presented in [7] and [8]. Further on, while the core of the N-max theorem is based on the zero-sum lemma, the shell is directly based on the regular n -simplex instead of indirectly using theorems based on similar structures. The suggested formulation and proof of the N-max theorem, which constitutes the proof of the optimality of N-max, is inspired by the minimax proposition presented in [9] and uses similar denotations and in some cases identical wording. The explicit incorporation of the Nash equilibrium in the proof of the N-max theorem is not required, but simplifies the process and could be interesting due to its common use in economics. A Nash equilibrium is briefly put a point ψ where no player has anything to gain by unilaterally changing its own strategy.

In the following equations, the simplified notation \max_{ξ_i} has been used to denote $\max_{\xi_i \in A_i}^i$ for any player i , operating solely on element i of the argument, where A_i denotes the strategies available for player i and according to the zero-sum lemma $\psi^* = \psi - \mu(\psi)$. The max function is here defined as an operator that considers the sequential order of the arguments such that given two equal values, only the first value in the sequence is regarded as a valid argument. The action ξ_1^* is here defined as the maximinimizer for player 1, if:

$$\max_{\xi_2} \max_{\xi_3} \dots \max_{\xi_N} \psi^*(\xi_1^*, \xi_2, \dots, \xi_N) \geq \max_{\xi_2} \max_{\xi_3} \dots \max_{\xi_N} \psi^*(\xi_1, \xi_2, \dots, \xi_N) \quad (21)$$

A maximinimizer for player i is thus an action that maximizes the payoff that player i can guarantee and solves the problem:

$$\max_{\xi_i} \max_{\xi_{i+1}} \dots \max_{\xi_N} \max_{\xi_1} \dots \max_{\xi_{i-2}} \max_{\xi_{i-1}} \psi^*(\xi_1, \xi_2, \dots, \xi_N) \quad (22)$$

This expression is simplified as:

$$\mathcal{N}_{j=i}^N \max_{\xi_j} \mathcal{N}_{j=1}^{i-1} \max_{\xi_j} \psi^*(\xi_1, \xi_2, \dots, \xi_N) \quad (23)$$

where \mathcal{N} is defined as a non-commutative operator for max and \mathcal{C} as its commutative counterpart. These two definitions are not conventional, but have been introduced to keep the mathematical expressions in this section as concise and clear as possible.

Theorem 1. Let $G = \langle \{1, 2, \dots, N\}, (A_i), (\psi_i^*) \rangle$ be a strictly competitive strategic game.

- (a) ξ_i^* is a maximinimizer for player i .
- (b) If (a) holds, then $(\xi_1^*, \xi_2^*, \dots, \xi_N^*)$ is a Nash equilibrium of G .

Proof. To first prove (a), let $(\xi_1^*, \xi_2^*, \dots, \xi_N^*)$ be an equilibrium of G . Then for player 1 and the basic sequence of moves, player 1 \rightarrow player 2 $\rightarrow \dots \rightarrow$ player N :

$$\begin{aligned} & \psi^*(\xi_1^*, \xi_2^*, \dots, \xi_{N-1}^*, \xi_N^*) = \\ & \max_{\xi_N} \psi^*(\xi_1^*, \xi_2^*, \dots, \xi_{N-1}^*, \xi_N) \geq \\ & \vdots \\ & \max_{\xi_2} \max_{\xi_3} \dots \max_{\xi_N} \psi^*(\xi_1^*, \xi_2, \dots, \xi_N) \geq \\ & \max_{\xi_1} \max_{\xi_2} \max_{\xi_3} \dots \max_{\xi_N} \psi^*(\xi_1, \xi_2, \dots, \xi_N) \end{aligned} \quad (24)$$

Thus:

$$\psi^*(\xi_1^*, \xi_2^*, \dots, \xi_N^*) \geq \mathcal{N}_{i=1}^N \max_{\xi_i} \psi^*(\xi_1, \xi_2, \dots, \xi_N) \quad (25)$$

The $N!$ possible sequences of moves give the general expression:

$$\psi^*(\xi_1^*, \xi_2^*, \dots, \xi_N^*) \geq \mathcal{C}_{i=1}^N \max_{\xi_i} \psi^*(\xi_1, \xi_2, \dots, \xi_N) \quad (26)$$

Since $t_i \cdot t_j < 0$ for $i \neq j$ and $2 \leq N < \infty$ (i.e. any positive change of the score for any of the players results in a negative change for all other players):

$$\psi^*(\xi_1^*, \xi_2^*, \dots, \xi_N^*) = \mathcal{C}_{i=1}^N \max_{\xi_i} \psi^*(\xi_1, \xi_2, \dots, \xi_N) \quad (27)$$

This gives that ξ_i^* is a maximinimizer for player i . To prove part (b), let:

$$\mathbf{v} = \mathcal{C}_{i=1}^N \max_{\xi_i} \psi^*(\xi_1, \xi_2, \dots, \xi_N) \quad (28)$$

where \mathbf{v} is according to the zero-sum lemma located in zero-sum space. Due to the symmetric properties of the regular n -simplex and the commutative properties of Eq. (28), hence $\mathbf{v}^* \geq \psi^*(\xi_1, \xi_2, \dots, \xi_i^*, \dots, \xi_N)$ for any i , thereby $\mathbf{v}^* = \psi^*(\xi_1^*, \xi_2^*, \dots, \xi_N^*)$. Thus \mathbf{v}^* is a Nash equilibrium of G . ■

It should be mentioned that there is a chance that there exists more than one equilibrium point if the max operator is defined as a non-sequential function that regards all equal values as equally valid. In the case of $N = 2$, it can be shown that all equilibria have the same payoffs [9]. In the case of $2 < N < \infty$ it is on the contrary relatively unlikely that two equilibrium points will coincide. It could however be argued that since multiple equilibria in general arise due to the simplistic nature of heuristic utility function model (for instance only using integer scores), if in theory the model would have been flawless, the chance would have been much smaller for multiple equilibria to occur, since the value of the utility function would have been closer to an irrational number than for instance a whole and the chance very small for any value to be repeated in one and the same game, unless the outcome of two or more sequences of actions would have resulted in exactly the same game position and score (given a decision tree that in reality is a simplification of a graph) which however would in similarity with the two-player case have yielded the same payoffs. In this context, game position is defined as the complete state of a game at any given moment. The definition of max in a sequential context is here thus not only a matter of practical implementation but is also reasonable from a pure theoretical point of view. The pseudocode for N-max is presented below:

```

NMax (node, p)
  if (leaf) return  $\psi - \mu(\psi)$ 
   $\alpha = -\infty$ 
  for each child
     $\psi^* = \text{NMax}(\text{child}, p^+)$ 
    if ( $\alpha < \psi_p^*$ )  $\alpha = \psi_p^*$ ,  $\psi_{\max}^* = \psi^*$ 
  return  $\psi_{\max}^*$ 

Initial call: NMax (start node, player)

```

IV. HYPERMAX

$$\alpha = [-\infty, -\infty, -\infty]$$

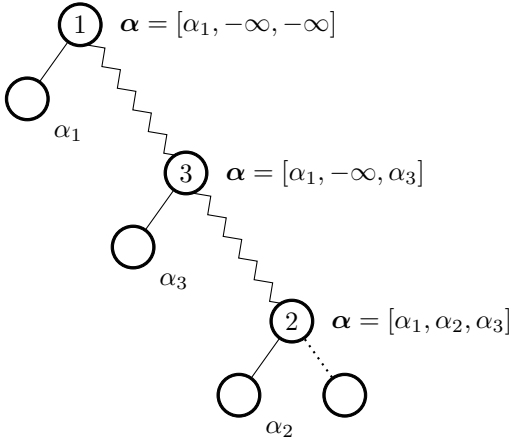


Figure 5. The approximation of deep alpha-beta pruning for N players presented in this paper requires at a minimum the establishment of a lower bound $\alpha_i > -\infty$ for each player i . The order by which the bounds are established does not matter. In this example, $N = 3$.

An approximated deep alpha-beta pruning method called hypermax [2], was derived for N-max based on Eq. (29) and Figs. 5-6. The general culling condition for this method with a set C_i denoting the boundary condition for player i may be expressed as:

$$\bigcap_{i=1}^N C_i = \emptyset \quad (29)$$

Henceforth, multiplayer alpha-beta pruning is simply referred to as *node culling* or *alpha-reduction* (since as mentioned before, in strategic games, what we often call decision trees are in reality decision graphs, why the term *reduction* could in this context be a more natural choice).

It is shown in [6] that there exists no computationally efficient alpha-reduction scheme for \max^N (and indirectly for N-max) comparable with the two-player case. The approximated alpha-reduction model eliminates nodes that do not have any chance to surface as the return value for N-max. In N-max, such nodes may however cause the elimination of the node associated with the return value of hypermax. As seen in Fig. 6, the approximated alpha-reduction condition for N-max may be expressed as $\alpha_1 > \mathbf{t}_1 \cdot \mathbf{x}$. Since \mathbf{t}_1 by definition is equal

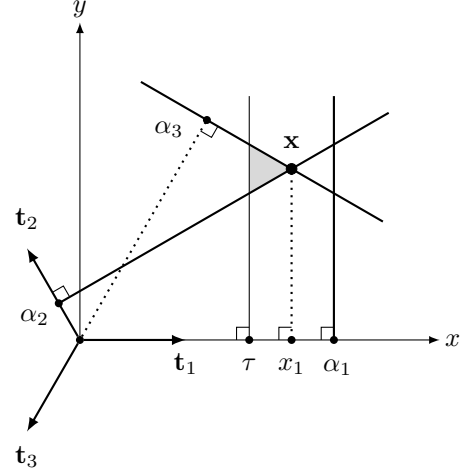


Figure 6. In this section x_1 is defined as the orthogonal projection of the n -tuple \mathbf{x} on the x -axis and \mathbf{x} as the intersection point of the hyperplanes perpendicular to the n column vectors $\{\mathbf{t}_2, \mathbf{t}_3, \dots, \mathbf{t}_N\}$ of the regular n -simplex matrix \mathbf{T} , where α_i is the shortest distance between hyperplane i and the origin. If $\alpha_1 > x_1$, the lower bounds α_i will eliminate the probability for any value to pass all bounds, thus all nodes are culled. If however $\alpha_1 = \tau$ (with $\tau \leq x_1$), there will in the general case exist a hypervolume inside the convex hull of the n -simplex where not all nodes can be culled. Thus for $\alpha_1 \leq x_1$ the condition for approximated alpha-reduction is not fulfilled. This figure depicts the special case of $N = 3$.

to the identity vector $\mathbf{i}_1 = [1 \ 0 \ 0 \ \dots \ 0]^T$, the culling condition may be simplified as $\alpha_1 > x_1$ where x_1 denotes the first element of \mathbf{x} , which can be calculated by the solution of \mathbf{x} with $\mathbf{b} = [\alpha_2 \ \alpha_3 \ \dots \ \alpha_N]^T$ in:

$$\mathbf{A} \cdot \mathbf{x} = [\mathbf{t}_2 \ \mathbf{t}_3 \ \dots \ \mathbf{t}_N]^T \cdot \mathbf{x} = \mathbf{b} \quad (30)$$

The equation for the general case (a hyperplane) may be derived by the dot product $\mathbf{n} \cdot \mathbf{x} = d$, where \mathbf{n} denotes a unit vector normal to the hyperplane and d is the shortest distance between the hyperplane and the origin $\mathbf{0}$. For the special cases of $N = 2$ and 3 , the zero-sum hyperplane is in this context reduced to a line in 2D versus a plane in 3D space. Since each row in \mathbf{A} represents the normal unit vector of the hyperplane associated with $d_i = \alpha_i$, the equation for a hyperplane i is equal to:

$$\sum_{j=1}^n t_{ji} \cdot x_j = \alpha_i \quad (31)$$

Eq. (30) is solved by $\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b}$, where \mathbf{A} for $n = 1$ and 2 is defined as:

$$\mathbf{T}_1 = [\mathbf{i}_1 \mid \mathbf{A}_1^T] = [1 \mid -1] \quad (32)$$

$$\mathbf{T}_2 = [\mathbf{i}_1 \mid \mathbf{A}_2^T] = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \quad (33)$$

For $N = 2$ and 3 the intersection points x_1 are given by:

$$^2x_1 = \mathbf{1}^T \cdot \mathbf{A}_1^{-1} \cdot \alpha_2 = 1 \cdot (-1) \cdot \alpha_2 = -\alpha_2 \quad (34)$$

$$^3x_1 = \mathbf{i}_1^T \cdot \mathbf{A}_2^{-1} \cdot \mathbf{b} = [1 \ 0] \begin{bmatrix} -1 & -1 \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix} =$$

$$\begin{bmatrix} -1 & -1 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \alpha_3 \end{bmatrix} = -\alpha_2 - \alpha_3 \quad (35)$$

With the addition of $\text{sum}(\alpha) = 0$, the culling conditions are thus equal to:

$$N = 2 : \alpha_1 + \alpha_2 \geq 0 \quad (36)$$

$$N = 3 : \alpha_1 + \alpha_2 + \alpha_3 \geq 0 \quad (37)$$

In the case of $N = 2$, the culling condition in Eq. (36) is identical to the one of standard two-player alpha-beta pruning with $\alpha_1 = \alpha$ and $\alpha_2 = -\beta$. The general proof that the

approximated culling condition is $\text{sum}(\alpha) \geq 0$ for $N \in \mathbb{N}_2$ is presented below.

Theorem 2. Given a regular n -simplex matrix \mathbf{T} with unit column vectors \mathbf{t}_i , the lower bounds $\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_N]^T$ and hyperplanes with normal unit vectors \mathbf{t}_i , each hyperplane placed at a shortest distance α_i from the origin, if $\text{sum}(\alpha) > 0$, no solution can satisfy the boundary conditions.

Proof. To start with, a general formula is derived for the calculation of $\mathbf{T} = [\mathbf{t}_1 \ \mathbf{t}_2 \ \dots \ \mathbf{t}_N]$ with $N \in \mathbb{N}_2$ and $N = n+1$. \mathbf{T} consists of unit column vectors, see Eq. (38), pointing from the origin and center of the regular n -simplex to its N vertices.

$$\mathbf{T} = \begin{bmatrix} 1 & \gamma_1 & \gamma_1 & \cdots & \gamma_1 & \gamma_1 & \gamma_1 & \gamma_1 \\ 0 & t_{22} & \gamma_2 & \cdots & \gamma_2 & \gamma_2 & \gamma_2 & \gamma_2 \\ 0 & 0 & t_{33} & \cdots & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & t_{(n-2)(n-2)} & \gamma_{n-2} & \gamma_{n-2} & \gamma_{n-2} \\ 0 & 0 & 0 & \cdots & 0 & t_{(n-1)(n-1)} & \gamma_{n-1} & \gamma_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & 0 & t_{nn} & \gamma_n \end{bmatrix} \quad (38)$$

The Eqs. (1) and (38) with $\gamma = \gamma_1 = -\frac{1}{n}$ give:

$$\left. \begin{array}{l} \mathbf{t}_1 \cdot \mathbf{t}_2 = 1 \cdot t_{12} = \gamma_1 \\ \mathbf{t}_1 \cdot \mathbf{t}_3 = 1 \cdot t_{13} = \gamma_1 \\ \vdots \\ \mathbf{t}_1 \cdot \mathbf{t}_{n+1} = 1 \cdot t_{1(n+1)} = \gamma_1 \end{array} \right\} \Rightarrow \gamma_1$$

$$\left. \begin{array}{l} \mathbf{t}_2 \cdot \mathbf{t}_3 = \gamma_1^2 + t_{22} \cdot t_{23} = \gamma_1 \\ \mathbf{t}_2 \cdot \mathbf{t}_4 = \gamma_1^2 + t_{22} \cdot t_{24} = \gamma_1 \\ \vdots \\ \mathbf{t}_2 \cdot \mathbf{t}_{n+1} = \gamma_1^2 + t_{22} \cdot t_{2(n+1)} = \gamma_1 \end{array} \right\} \Rightarrow \gamma_2$$

$$\left. \begin{array}{l} \mathbf{t}_{n-1} \cdot \mathbf{t}_n = \gamma_1^2 + \gamma_2^2 + \dots + \gamma_{n-2}^2 \\ \quad + t_{(n-1)(n-1)} \cdot t_{(n-1)n} = \gamma_1 \\ \mathbf{t}_{n-1} \cdot \mathbf{t}_{n+1} = \gamma_1^2 + \gamma_2^2 + \dots + \gamma_{n-2}^2 \\ \quad + t_{(n-1)(n-1)} \cdot t_{(n-1)(n+1)} = \gamma_1 \end{array} \right\} \Rightarrow \gamma_{n-1}$$

where:

$$\gamma_1 = t_{12} = t_{13} = \dots = t_{1(n+1)} \quad (39)$$

$$\gamma_2 = t_{23} = t_{24} = \dots = t_{2(n+1)} \quad (40)$$

$$\gamma_{n-1} = t_{(n-1)n} = t_{(n-1)(n+1)} \quad (41)$$

$$\begin{aligned} \mathbf{t}_n \cdot \mathbf{t}_{n+1} &= \gamma_1^2 + \gamma_2^2 + \dots + \gamma_{(n-1)(n-1)}^2 \\ &\quad + t_{nn} \cdot t_{n(n+1)} = \gamma_1 \end{aligned} \quad (42)$$

Since \mathbf{t}_i is a unit vector:

$$\gamma_1^2 + t_{22}^2 = 1 \quad (43)$$

$$\gamma_1^2 + \gamma_2^2 + t_{33}^2 = 1 \quad (44)$$

$$\vdots$$

$$\gamma_1^2 + \gamma_2^2 + \dots + \gamma_{(n-1)}^2 + t_{nn}^2 = 1 \quad (45)$$

$$\gamma_1^2 + \gamma_2^2 + \dots + \gamma_{(n-1)}^2 + t_{n(n+1)}^2 = 1 \quad (46)$$

The Eqs. (45)-(46) give:

$$t_{nn}^2 = t_{n(n+1)}^2 \quad (47)$$

As \mathbf{t}_n and \mathbf{t}_{n+1} cannot be parallel, Eq. (47) gives that $\gamma_n = t_{n(n+1)} = -t_{nn}$. The Eqs. (1)-(2) and (38)-(47) yield thereby the following algorithm for the recursive calculation of \mathbf{T} :

$$\left. \begin{array}{l} t_{ii} = \sqrt{1 - \sum_{j=1}^{i-1} \gamma_j^2} \\ \gamma_i = -\frac{t_{ii}}{n+1-i} \end{array} \right\} 1 \leq i \leq n \quad (48)$$

The definition of matrix inverse $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, yields:

$$\mathbf{w}^T \cdot \mathbf{A} = \mathbf{i}_1^T \quad (49)$$

where \mathbf{w} is the first row of \mathbf{A}^{-1} , which we assume consists of the elements $w_k = -1$ for $1 \leq k \leq n$. As $(\mathbf{A}\mathbf{B})^T = \mathbf{B}^T\mathbf{A}^T$ for any matrices \mathbf{A} and \mathbf{B} of appropriate size, $\mathbf{A}^T \cdot \mathbf{w} = \mathbf{i}_1$, or:

$$\begin{bmatrix} \gamma_1 & \gamma_1 & \gamma_1 & \dots & \gamma_1 & \gamma_1 & \gamma_1 \\ t_{22} & \gamma_2 & \gamma_2 & \dots & \gamma_2 & \gamma_2 & \gamma_2 \\ 0 & t_{33} & \gamma_3 & \dots & \gamma_3 & \gamma_3 & \gamma_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \gamma_{n-2} & \gamma_{n-2} & \gamma_{n-2} \\ 0 & 0 & 0 & \dots & t_{(n-1)(n-1)} & \gamma_{n-1} & \gamma_{n-1} \\ 0 & 0 & 0 & \dots & 0 & t_{nn} & \gamma_n \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{n-2} \\ w_{n-1} \\ w_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (50)$$

The Eqs. (48) and (50) give the relations:

$$t_{kk} = -(n+1-k)\gamma_k \quad (51)$$

$$t_{kk} \cdot w_{k-1} + \gamma_k \sum_{j=k}^n w_j = 0 \quad (52)$$

which combined with $\gamma_k \neq 0$ gives:

$$w_{k-1} = \frac{1}{n+1-k} \sum_{j=k}^n w_j \quad (53)$$

Solving Eq. (53) recursively, starting with $k = n$ and decrementing k by one for each step yields:

$$\begin{aligned} w_{n-1} &= w_n \\ w_{n-2} &= \frac{1}{2}(w_{n-1} + w_n) = w_n \\ w_{n-3} &= \frac{1}{3}(w_{n-2} + w_{n-1} + w_n) = w_n \\ &\vdots \\ w_1 &= \frac{1}{n-1}(w_2 + w_3 + \dots + w_n) \\ &= \frac{1}{n-1}((n-1)w_n) = w_n \end{aligned} \quad (54)$$

The substitution of w_1, w_2, \dots, w_n by w and the multiplication of the first row of \mathbf{A}^T with \mathbf{w} gives finally the relation $n\gamma_1 w = 1$, which with $\gamma = \gamma_1 = -\frac{1}{n}$ confirms that $w_k = -1$ for $k \in \{1, 2, \dots, n\}$ and therefore that the orthogonal projection of the crossing point \mathbf{x} on the x -axis is equal to $x_1 = -\sum_{i=2}^N \alpha_i$. Combined with $\alpha_1 > x_1$, this yields that for $N \in \mathbb{N}_2$, the condition for the elimination of any solution that can satisfy all boundary conditions is equal to:

$$\sum_{i=1}^N \alpha_i > 0 \quad (55)$$

■

Any individual scaling of the columns \mathbf{t}_i in \mathbf{T} by a factor κ_i generalizes Eq. (55) into $\kappa \cdot \alpha > 0$. This relation remains therefore intact if \mathbf{T} is rescaled by a constant non-zero real value $\kappa_i = c$. The replacement of \mathbf{T} by $\hat{\mathbf{T}}$ does therefore not affect Eq. (55). In practical applications, $\text{sum}(\alpha) = 0$

is added to the pruning condition to minimize the number of computations. The pseudocode for hypermax with player scores $\psi \in \mathbb{R}^N$ is presented below:

```
Hypermax (node, p,  $\alpha$ )
  if (leaf) return  $\psi - \mu(\psi)$ 
  for each child
     $\psi^* = \text{Hypermax}(\text{child}, p^+, \alpha)$ 
    if (first-child)  $\psi_{\max}^* = \psi^*$ 
    if ( $\alpha_p < \psi_p^*$ )  $\alpha_p = \psi_p^*$ ,  $\psi_{\max}^* = \psi^*$ 
    if ( $\text{sum}(\alpha) \geq 0$ ) break
  return  $\psi_{\max}^*$ 
```

Initial call: Hypermax (start node, player, $[-\infty -\infty \dots -\infty]^T$)

Hypermax is in this context considered to be an optimal mathematical extension of alpha-beta pruning in the sense that it extends the method to multiplayer games in the most straightforward fashion.

Coalition building for N-max and hypermax may be divided into the categories weak and strong. For weak coalitions, a graph connectivity matrix \mathbf{C} such that $\psi \leftarrow \mathbf{C} \cdot \psi$ was in this work found useful for the definition of the relationship between the players. For strong coalitions, players forming an alliance could be reduced into a single player having access to different sets of game assets for each move.

As a final note, if the utility function for N-max or hypermax is for any specific game better off measuring the score relatively, ψ^* could be normalized after calculation.

V. EXPERIMENTAL RESULTS

In a case study, using a four-handed computer chess program, it was shown that a difference in win counts between max^N and N-max could statistically be established, as expected, based on a very few experiments.

The hypermax algorithm was further tested in simulations measuring the execution speed and in a few case studies measuring the gameplay quality. The simulations were based on a random number generator, such that the score of each player i for each node was calculated by $\psi_i = \psi_i^{\text{parent}} + \text{Rand}()$, where $\text{Rand}()$ is a random number generator with approximately a uniform probability distribution symmetric around zero. In Table I, the input parameters are the number of players N , number of children b per node and search depth d . A_0 and A_{best} denote the number of nodes traversed by N-max (exhaustive search) versus the best case when the pruning condition $\text{sum}(\alpha) \geq 0$ is always satisfied. In Table I, $\mu = A_{\text{ave}}/A_{\text{best}}$ without sorting (where A_{ave} denotes average

N	b	d	A_{best}	A_0/A_{best}	u_1	u_2	μ	μ_s
3	4	6	1249	4.4	4.88	0.273	1.75	1.23
3	4	9	21556	16.2	5.26	0.253	1.75	1.23
3	4	12	351607	63.6	5.37	0.249	1.82	1.20
3	4	15	5653114	253.3	5.39	0.247	1.56	1.12
3	4	18	90560125	1011.8	5.40	0.247	1.30	1.05
3	4	21	1449404032	4045.8	5.40	0.247	1.16	1.02
3	8	6	18333	16.3	4.48	0.255	1.77	1.17
3	8	9	1198816	128.0	4.57	0.250	1.48	1.08
3	8	12	76932323	1020.9	4.59	0.249	1.19	1.04
3	16	6	278197	64.3	4.24	0.251	1.78	1.08
3	16	9	71621560	1023.4	4.27	0.250	1.20	1.02
3	32	6	4324197	256.3	4.12	0.250	1.36	1.07
4	4	8	24120	3.6	5.89	0.226	1.49	1.13
4	4	12	1697044	13.2	6.47	0.206	1.40	1.10
4	4	16	111227024	51.5	6.63	0.201	1.21	1.03
4	4	20	7161077388	204.7	6.67	0.200	1.09	1.01
4	8	8	1452180	13.2	5.54	0.206	1.40	1.08
4	8	12	764214800	102.8	5.69	0.201	1.12	1.02
4	16	8	88785324	51.6	5.29	0.202	1.19	1.04
5	4	10	448943	3.1	6.85	0.195	1.35	1.09
5	4	15	129130324	11.1	7.70	0.173	1.20	1.04
5	8	10	110644503	11.1	6.59	0.173	1.23	1.07

Table I

NODE COUNTS FOR EXHAUSTIVE SEARCH (A_0) VERSUS HYPERMAX (A_{AVE} AND A_{BEST}) WITH $\mu = A_{\text{AVE}}/A_{\text{BEST}}$ BASED ON 8 SIMULATIONS PER LINE

node counts) and μ_s with sorting (in reverse order with respect to ψ^*) of the children for each node prior to the evaluation of the pruning condition. By the estimation of the proportional coefficients u_1 and u_2 in Eqs. (56)-(57), the algorithm showed for practical applications and the best case as expected by [12] to be $O(b^{d(N-1)/N})$.

$$A_{\text{best}} = u_1 \cdot b^{d(N-1)/N} \quad (56)$$

$$\frac{A_0}{A_{\text{best}}} = u_2 \cdot b^{\frac{d}{N}} \quad (57)$$

The conclusion is thus that the average complexity of the branching factor for hypermax is equal to its best case $O(b^{d(N-1)/N})$.

The relative enhancement factor κ_α for an algorithm was in this work defined as the increase in efficiency when given a node count $A_\alpha = A_{\text{ave}}$, the computational power is increased by a factor κ . Since the adaption of N-max to parallel computing is straightforward and the number of nodes in hypermax that are not affected by the pruning scheme is $O(b^{d(N-1)/N})$, the adaption of hypermax to parallel computing is straightforward. Given the relation in Eq. (56) for a constant value u_1 and the relations $A_0 \propto b^d$ and $A_\alpha \propto A_{\text{best}}$, the relative enhancement factor for hypermax is equal to:

$$\kappa_\alpha = \frac{b^{\frac{N}{N-1} \frac{\log(\kappa A_\alpha)}{\log b}}}{\kappa A_\alpha} \bigg/ \frac{b^{\frac{N}{N-1} \frac{\log A_\alpha}{\log b}}}{A_\alpha} = \kappa^{\frac{1}{N-1}} \quad (58)$$

Eq. (58) is an approximation since u_1 and the relations $A_0 \propto b^d$ and $A_\alpha \propto A_{\text{best}}$ are approximations.

For shallow trees and a low branching factor, hypermax returned the same value as N-max by a relatively high probability, see Table II. As a note, normalization of ψ^* showed to increase μ and μ_s in Table I (with μ in some cases significantly

more than μ_s) and the frequency values of Table II. As an example $\mu_s = 1.57$ for $N = 3$, $b = 4$, $d = 21$ and $\mu_s = 1.10$ for $N = 5$, $b = 8$, $d = 10$. In Table II for $d = 10$, the frequencies after normalization of ψ^* were counted to 195 and 454, compared with 126 versus 289, for 1000 trees per simulation.

In a case study, by counting wins in four-handed computer chess, it was not possible given the time constraints and the allocated computational power (of approximately 100 GFLOPS) to find a depth limit where hypermax could statistically be proven to part from N-max.

This indicates that the return values of hypermax that are not identical with the return values of N-max are (for limited search depths) in average still useful approximations. Such limit as a function of N and b is proposed to be explored in future work.

VI. CONCLUSION

N-max, a redefinition of \max^N , extends minimax to N -player games without compromising the zero-sum properties of minimax. Hypermax, an extension of alpha-beta pruning to N -player games showed in preliminary experiments to efficiently optimize N-max. During the derivation of the zero-sum lemma, a new elementary method was further found for the calculation of the dihedral angle of the regular n -simplex.

REFERENCES

- [1] P. Ciancarini and G. P. Favini, "Monte Carlo Tree Search in Kriegspiel", *Artificial Intelligence*, vol. 174, no. 11, pp. 670-684, 2010.
- [2] M. Fridenfolk, *Method for Optimal N-Person Extension of Minimax and Alpha-Beta Pruning*, Patent Pending, no. SE1230120-6, November, 2012.

d	3	4	5	6	7	8	9	10
None	963	871	753	542	393	284	199	126
Sort	1000	944	892	804	640	501	383	289

Table II

FREQUENCY OF IDENTICAL RETURN VALUES FOR N-MAX AND HYPERMAX FOR 1000 TREES WITH $N = 3$ AND $b = 4$ AS A FUNCTION OF SEARCH DEPTH

- [3] S. Gelly and D. Silver, "Monte-Carlo Tree Search and Rapid Action Value Estimation in Computer Go", *Artificial Intelligence*, vol. 175, no. 11, pp. 1856-1875, 2011.
- [4] F. H. Hsu, "Cracking Go", *IEEE Spectrum*, vol. 44, no. 10, pp. 50-55, 2007.
- [5] D. E. Knuth and R. W. Moore, "An Analysis of Alpha-Beta Pruning", *Artificial Intelligence*, vol. 6, pp. 293-326, 1975.
- [6] R. E. Korf, "Multiplayer Alpha-Beta Pruning", *Artificial Intelligence*, vol. 48, no. 1, pp. 99-111, 1991.
- [7] C. A. Luckhardt and K. B. Irani, "An Algorithmic Solution of N-Person Games", In *Proceedings AAAI-86*, vol. 6, pp. 158-162, Philadelphia, PA, 1986.
- [8] J. F. Nash, "Equilibrium Points in N-Person Games", *PNAS*, vol. 36, no. 1, pp. 48-49, 1950.
- [9] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT, Cambridge, MA, 1994.
- [10] H. R. Parks and D. C. Wills, "An Elementary Calculation of the Dihedral Angle of the Regular n -Simplex", *The American Mathematical Monthly*, vol. 109, no. 8, pp. 756-758, 2002.
- [11] N. R. Sturtevant, "Last-Branch and Speculative Pruning Algorithms for Maxⁿ", *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 18, pp. 669-678, 2003.
- [12] N. R. Sturtevant and R. E. Korf, "On Pruning Techniques for Multi-Player Games", In *Proceedings of The National Conference on Artificial Intelligence (AAAI)*, pp. 201-207, 2000.
- [13] J. von Neumann, "Zur Theorie der Gesellschaftsspiele", *Mathematische Annalen*, vol. 100, pp. 295-320, 1928.
- [14] D. C. Wills, *Connections Between Combinatorics of Permutations and Algorithms and Geometry*, PhD Thesis, Oregon State University, 2009.