Empirical Game-Theoretic Analysis of Chaturanga

Christopher Kiekintveld, Michael P. Wellman, and Satinder Singh

University of Michigan
Computer Science and Engineering
Ann Arbor, MI 48109-2121 USA
{ckiekint,wellman,baveja}@umich.edu

ABSTRACT

We analyze 4-player chaturanga (an ancient variant of chess) using the methods of empirical game theory. Like chess, this game is computationally challenging due to an extremely large strategy space. From the perspective of game theory, it is more interesting than chess because it has more than 2 players. Removing the 2-player restriction allows multiple equilibria and other complex strategic interactions that require the full tool set of game theory. The major challenge for applying game theoretic methods to such a large game is to identify a tractable subset of the game for detailed analysis that captures the essence of the strategic interactions. We argue that the notion of strategic independence holds significant promise for scaling game theory to large games. We present preliminary results based on data from two sets of strategies for chaturanga. These results suggest that strategic independence is present in chaturanga, and demonstrate some possible ways to exploit it.

1. INTRODUCTION

For many real games of interest, the underlying game form is not directly known by the players and/or analyst. Instead, various kinds of evidence may be available for estimating the game (for example, simulation). We refer to techniques for analyzing such games as *empirical game theory*. A distinguishing feature of empirical game theory is the direct emphasis on the process of discovering the parameters of the game. This is distinct from learning about opponents' strategies during repeated interactions, though the two types of learning are sometimes conflated. These scenarios can be modeled as games of incomplete information with uncertain payoffs [3]. However, explicitly modeling the process of discovering the game results in analytically intractable models, even for relatively simple scenarios.

Uncertainty about the game form may arise from stochastic payoffs or computational complexity. First, consider the case where payoffs are drawn from an unknown distribution but players can gather samples individually. To model this as a game of incomplete information, we would need to define player types based on the possible combinations of samples and compute the necessary conditional belief distributions for each type. This type space grows exponentially in the number of samples. If players make choices about which profiles are sampled or how many samples are taken, the model becomes even more complex. This difficulty can arise even in games with small numbers of players and actions.

Uncertainty also arises from computational complexity (bounded rationality). There may simply be too many outcomes to compute, represent, and reason about. This is common when games have sequential interactions, as in chess or go. Sequences of moves result in a combinatorial explosion in the number of possible outcomes. It may be relatively easy to determine the outcome for a particular sequence, but players can compute and store the outcomes for only a small fraction of the possible sequences of play. In addition, players may be unaware of which outcomes opponents have explored. This difficulty can arise even in deterministic games. Obviously, these two forms of uncertainty are not mutually exclusive; large stochastic games are also common.

We discuss preliminary results from our investigation of 4-player chaturanga, an ancient variant of chess. Like chess, this game is challenging because of the massive number of possible sequences of play. This variant was selected specifically because it has more than two players, and most computationally challenging games that have received extensive attention (e.g., chess, checkers, backgammon, go) belong to the restricted class of two-player, zero-sum games. The move to more than two players raises important strategic issues, including the potential for multiple equilibria and alliances between players.

We employ methods from empirical game theory to analyze chaturanga, an approach that takes seriously the potential for complex strategic interactions in the game. We argue that methods for exploiting structure are likely to be critical for success in analyzing large games, just as they are for large single-agent learning problems. A common approach in single-agent problems is dimensionality reduction, where the goal is to create a simpler problem by ignoring irrelevant information. In games, we can also simplify the problem by identifying *strategically independent* choices, in addition to payoff-irrelevant choices. In other words, it may be possible to optimize some choices without regard to the strategies selected by opponents, obviating the need to gather data on the outcomes for many strategy profiles. We present evidence that some form of strategic independence exists in chaturanga, and demonstrate some potential ways of exploiting this.

2. ANALYSIS OF LARGE GAMES

It is impossible to perform complete game-theoretic analysis on games with very large strategy spaces. *Heuristic strategy analysis* deals with this problem by analyzing a heuristically chosen subset of the strategy space[18]. Even though this type of analysis is necessarily incomplete, it can be very useful in practice. It has been

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

applied to perform principled analysis of strategies in common use and proposed alternatives [19, 11, 8] and to mechanism design [17]. A particularly interesting variation is to iterate this analysis, using directed search to generate new candidate strategies. Phelps et al. [10] use this approach coupled with a genetic algorithm for search to investigate bidding strategies for double auctions. We identify three basic steps for empirical analysis of a game:

1. Identify candidate strategies
2. Generate an empirical payoff matrix
3. Analyze the resulting game

There are many alternatives for implementing each step in this process, and they are only beginning to be explored to determine the most effective approaches. These basic steps can also be iterated in various ways, potentially improving the quality of the analysis. In general, we do not expect to be able to identify a uniquely compelling strategy using this approach. However, many of the initial studies listed above have been able to draw useful and sometimes surprisingly strong conclusions.

There is a tradeoff when constraining the set of strategies under consideration. Including more strategies increases the computational costs, but strengthens the results. One way to mitigate this problem is to employ methods for reasoning based on incomplete game specifications. We can verify the stability of individual profiles by examining only the one-step deviations from this profile. Even if we know the payoffs for only a subset of the deviations, we can still compute a lower bound on the maximum benefit to deviating (ϵ). Using this observation, we can implement a simple best-first search procedure for finding stable profiles (ϵ-Nash equilibria). Given any partial data set, always sample an unexplored deviation from a profile in the data set with the lowest known bound on ϵ. It is likely that this search can be improved using additional heuristics, but it has already shown promise in applications [20].

3. STRATEGIC INDEPENDENCE

Game theory offers powerful tools for studying situations where a player's choice of action depends on the choices made by other players. However, full game-theoretic analysis does not scale well to large games because the number of possible joint strategy profiles grows very rapidly. Exploiting assumptions about strategic independence is one potential way to scale game theory up to larger games. Conceptually, we would like to focus game-theoretic analysis on the most important strategic interactions between players, using single-agent methods to optimize non-strategic choices. Formalisms like graphical games [4] and multi-agent influence diagrams (MAIDs) [5] provide structured representations exploiting strategic independence. These representations are more compact and provide considerable computational advantages in computing solutions. Exploiting strategic independence in empirical games potentially offers even greater advantages, since it could dramatically reduce the number of samples needed to create a useful representation for a game. This in turn reduces the need for game theory practitioners to make arbitrary modeling assumptions to achieve tractable models. The challenge is to find ways of identifying and exploiting these relationships based on the available data (i.e., to infer the graph structure taken as given in existing work).

A very common way of assessing performance in complex games is to assign strategies[1] a rank ordering or numeric rating. A wide variety of systems exist, and they are used for most major competitive events [14, 9], including chess [2]. The appeal of rankings systems is obvious. They are are useful, conceptually straightforward, and can produce reasonable results based on even sparse data. To achieve these properties they make very strong assumptions. A perfect ranking requires a complete, strict, and transitive dominance relationship between the strategies. This is typically motivated by assuming that each strategy has an underlying strength (or strength distribution, to allow for stochasticity), independent of the opponent's strategy. The winner of a game is determined by comparing the strengths of the strategies. Much of the work on ranking systems considers different methods for estimating the strengths of strategies based on paired comparisons.

We view the assumption that strategies can be ordered according to an underlying strength parameter as a particularly strong form of strategic independence assumption. Not only does it assume that all of the complexities of the interactions between player strategies can be compressed into a single "strength" dimension, it also assumes a very simple best-response function. It is always better to play a stronger strategy, regardless of the opponent's strategy. The disadvantage of these assumptions is that rankings are not powerful enough to express tradeoffs between strategies that are dependent on the opponent's strategy.

Game theory and rankings are at opposite ends of a spectrum. Game theory assumes that all strategic choices interact, while rankings assume essentially no interactions. Game theory can potentially make more accurate predictions, but does not scale well to large games. Rankings are simpler and require much less data, but do not have the power of game theory. Of course, these are not the only alternatives. We believe that there are many interesting opportunities to bridge the gap between these two extremes and combine the advantages of both by varying the assumptions made about strategic independence. The fact that rankings enjoy such popularity suggests that many games of interest do have aspects of strategic independence.

In addition to the basic questions of whether or not strategic independence exists in games of interest and what forms it takes, there are also many interesting possibilities for how to exploit it. One is to explicitly try to derive the structure of the dependence relationships from the data (a problem akin to structure discovery problems for Bayesian networks). Other methods might exploit independence less directly. For instance, we could use broad measures of strategy performance as a heuristic for directing a search in profile space for stable profiles. The process of iterating local search to generate candidate strategies with game-theoretic analysis of candidates also exploits strategic independence. The local search procedures ignore strategic considerations, but should quickly optimize independent aspects of strategy performance. Strategic independence could also be used to guide the development of a strategy space for experimentation. For instance, we can explicitly adopt structured strategy representations and identify variables likely to be strategically independent.

4. 4-PLAYER CHATURANGA

Chaturanga is an ancient variant of chess from India. We consider a 4-player version of the game. The game is played on the same board as standard chess, with a modified configuration to allow for four players. The starting configuration is shown in Figure 1. Each player has eight pieces: 1 king, 1 rook, 1 knight, 1 boat, and 4 pawns. All pieces except the boat are identical to pieces in standard chess. The boat moves diagonally in any direction, but can only move to the second square; it must jump over exactly 1 space. Pawns may advance only in the direction they are facing, and can

[1] Alternatively, players or teams. We ignore this potential distinction for our purposes.

Figure 1: The game board for chaturanga. The simulator and visualization were developed by Cougaar Software.

promote to any other piece type except king when they reach the 8th rank. There is no initial double move for pawns. There is no notion of check or mate,[2] so kings must be taken to eliminate a player. The pieces of eliminated players remain on the board as obstacles, but cannot move. The game ends when a single king remains, an identical position occurs three times during the game, or each player has made at least 50 consecutive moves without moving a pawn or capturing a piece.[3] One point is divided equally between all players with kings remaining on the board when the game ends; any player without a king gets a score of 0.

A *strategy* for chaturanga defines a move for each possible board configuration. An *outcome* is a legal sequence of moves that constitutes a game. Like chess, chaturanga is challenging primarily because of the combinatorial explosion in the number of possible sequences of moves and resulting board configurations. This presents problems both for representing strategies and reasoning about the outcomes, since complete representations (e.g., tables) are orders of magnitude too large to even store in a modern computer. In other ways, chaturanga is a relatively simple game: it is deterministic, has no hidden state during game play, and has a very limited set of possible payoffs. It is also relatively easy to simulate a particular instance of the game to determine payoffs.

5. CHATURANGA STRATEGIES

From the viewpoint of empirical game theory, we can call any process that selects a move for each possible board position a strategy; the details are not important. The raw strategy space is much too large to reason about directly, so we define a transformed strategy space for analysis. This space defines a strategy based on possible parameter settings for a move selection algorithm that combines a feature-based evaluation function with online minimax search. We have also implemented a reinforcement learning algorithm for performing local search to estimate an evaluation function from sample games. Similar methods have been used extensively for playing chess and other games; two examples are the KnightCAP chess agent [1] and Samuel's checkers agent [13]. There are some challenges that arise in extending these techniques to games with more than two players, but the primary novel idea we explore is the idea of using game-theoretic analysis to determine parameter settings.

5.1 The Evaluation Function

The evaluation function provides an estimate of the expected payoffs for each player in the game for any legal board position. It is very important to understand that the "correct" evaluation function depends on the strategies employed by all players, so the function represents predictions about both the strategies opponents will play and the likely outcomes. These predictions depend on both the player's knowledge of the game and assumptions about opponents (e.g., rationality). The evaluation function is parameterized, and different settings allow the function to represent a wide range of beliefs (though clearly not all possible beliefs). We explore various methods for setting the parameters based on combinations of local search and game theory.

The representation is based on features of the game board that consider factors like material value, threats, protections, board control, king mobility, etc. These features are combined to form a measure of the raw position strength of each player, which we denote S_i^{pos}. The function estimates separately the value of each piece, $V(\cdot)$, and threat level, $T(\cdot)$, for each piece. The threat level corresponds roughly to the probability that the piece will be taken in the near future. $V(\cdot)$ and $T(\cdot)$ are both determined by weighted linear combinations of the relevant features. These weights are parameters of the evaluation function, with a separate weight for each piece type. King values and threats are treated separately because of the special importance of the king (K). In addition to the values ascribed to the individual pieces we allow for an emergent value EV. All of the threats and values are combined using the functional form

$$S_i^{pos} = T(K)\left[\sum_{p\in\{pieces\setminus K\}} T(p)V(p) + V(K) + EV\right]. \quad (1)$$

In 2-player games it is straightforward to compare the position strengths of the two players and translate this into an expected outcome; in chess an advantage equivalent to one pawn usually evaluates to a near-certain win. For a 4-player game the situation is more complex, since there are 4 strengths to consider and they need not be weighted equally. We must also account for the nonlinear relationship between differences in position strength and expected outcomes. Our evaluation function performs pairwise comparisons using sigmoid functions to determine a score for each pair of players:

$$PW_{i,j} = \frac{1}{1 + e^{-m(S_i^{pos} - S_j^{pos}))}}. \quad (2)$$

The comparisons use a multiplier m that depends on the relationship between the players, allowing differences in positional strength relative to different players to have different weightings in the final evaluation. For instance, we have parameters that vary m depending on the relative position of the two players on the board (right, left, or diagonal) and the rank ordering of the raw strength scores. The multiplier allows players to "gang up" on certain opponents. Finally, these pairwise comparisons are combined and normalized to yield a predicted outcome for each player:

[2]Taking the king might not be the best move, since other players remain alive after the king is taken.

[3]Note that these stopping conditions guarantee that games are finite.

$$S_i = \prod_{j \in \{players \setminus i\}} PW_{i,j}. \quad (3)$$

The motivation for taking the product of the pairwise scores is that this would represent the joint probability of winning all of the pairwise match-ups if these were independent events and the pairwise scores are interpreted as probabilities. The final scores are normalized to sum to 1 so that they can be interpreted as expected scores. This interpretation allows us to estimate the parameters of the evaluation function directly from simulation data.

5.2 Learning the Evaluation Function

We use reinforcement learning [16] to approximate evaluation functions from simulation data. The algorithm uses TD(λ) updates with numeric approximations of the error gradient with respect to the parameters of the evaluation function. We do not currently use leaf-node updating as suggested by Baxter et al. [1], partially because we often run simulations with very shallow online search. All of the standard caveats of reinforcement learning with function approximation apply in this case; in particular, the learning process is not guaranteed to converge on a reasonable value function.

It is important to note that the resulting evaluation function may be highly dependent on the specific profiles of agents in the pool of training data. A common way of generating reasonable opponents is to use self-play, where the strategy is continually updated and pitted against itself. There are other alternatives including play against human experts[4] and play against a larger sampling of "old" opponents [12]. None of these solves the potential problem of multiple stable profiles. For this reason, we view learning as a local search procedure that should be augmented with more detailed game-theoretic reasoning.

5.3 Online Minimax Search

Online minimax search is used by virtually all successful chess agents to improve evaluations of potential moves. Exactly why exactly minimax search improves evaluations is not fully understood [7], but it works extremely well in practice. The standard version of minimax search applies to 2-player games, but there is a straightforward extension to N-player games known as $\max^N$ [6]. A thorough discussion of solution algorithms, pruning methods, and other issues can be found in [15]. For a variety of reasons, we find deep search to be a less compelling approach to solving N-player games. One problem is that standard pruning techniques are much less effective at reducing the search space for N-player games. Another is that the minimax rule may not yield a unique prediction for play in the game tree, since players may need to select between moves that they are indifferent between but affect the stakes of the other players differently. There are also more opportunities for making prediction errors, since each player must predict more opponent moves that are unlikely to be selected using the same evaluation function.

Despite these caveats, online search is still likely to be useful to some degree in N-player games. We have implemented several search variants including depth-first search, iterative deepening with a time limit, and beam search. The beam search expands nodes selectively, searching only the most promising paths at each level of the tree. This search can expand either a fixed number of the top nodes, or all nodes within a fixed value of the maximum. Beam search is deeper that DFS, but risks ignoring useful lines of play. The specific search methods used are controlled by parameters, which we view as extensions of the possible strategy space. Regardless of the method used, search is computationally demanding. To reduce the time required to run simulations, we often restrict the strategy space we consider to strategies with very limited or no search capabilities.

[4]If they are available. We do not know of any experts in 4-player chaturanga.

6. EXPERIMENTS

We present preliminary data we have gathered on two strategy spaces in chaturanga. While not originally designed for this purpose, analysis of these experiments provides suggestive evidence about the existence of strategic independence in chaturanga and potential approaches for exploiting it. The first hypothesis we examine is that the average score for each strategy across all profiles is correlated with the likelihood that the strategy appears in stable (low ϵ) profiles. This prediction is based on the concept of an underlying strength associated with each strategy. The second hypothesis is that strategies learned using reinforcement learning should perform well (appear in stable profiles) even against novel strategies; in other words, learning should generalize. This requires that at least some aspects of a good strategy can be learned without regard for the opposing strategies. The final hypothesis is that a strategy space based on a single high-level parameter of interest (selected by hand) should have interesting strategic interactions (no obvious stable profiles or dominant strategies). This hypothesis bears on the need for game-theoretic reasoning in this domain.

For these experiments we collected data by sampling the payoffs for profiles of strategies using a simulator. Strategies are defined by a vector of parameter settings for the evaluation function and online search. We introduce a small amount of noise into each strategy by forcing the agent to select a random move some percentage of the time (e.g., 5%). Even a small amount of noise introduces significant variation in the games played, since all moves after a random move are likely to be different. We can think of this as transforming each strategy into a mixture of similar strategies, presumably making results less dependent on the exact details of the particular strategies selected.

The first strategy space contains 17 strategies with a broad range of evaluation functions and online search settings. These strategies are listed in Table 1. We refer to this as the "exploratory" strategy space. There are a total of $17^4 = 83521$ profiles in this strategy space, and we have collected 30 payoff samples for 7863 of these profiles, approximately 9% of the total profile space. The data was collected using a version of the best-first search procedure described in Section 2.

The second strategy space was designed to emphasize high-level strategic interactions between the players. All strategies in this space use the same parameter settings for most of the evaluation function weights. These settings are the same as the learned evaluation function for strategy 7 in the previous strategy space. However, the strategies differ in how they weight pairwise differences in positional strength with the other players depending on the spatial orientation of the players. For purposes of easy visualization and demonstration, we project this weighting into a single "attack angle" parameter. We gathered data for 9 discrete instantiations of this parameter, plus 1 additional strategy that weights the players evenly. These strategies are shown in Figure 2. A small weight is places on all opponents by default. The attack angle parameter distributes the majority of the weight between the opponents it is pointing at, either 100% for a single agent or 33% and 66% split between two opponents.

For this 10-strategy game we gathered data for the entire space of 10000 profiles, collecting at least 50 samples for each profile. To

Table 1: **Strategies in the learning/search strategy space. A search ply represents a single move by a single player.**

Strategy	Description
0	**Random.** Selects uniformly at random from available moves.
1	**Hand-set 1.** Hand-set weights.
2	**Hand-set 2.** Hand-set weights.
3	**Material only.** Weights for material (pieces) only.
4	**Learn 1.** Learned from approximately 1000 games of self-play.
5	**Learn 2.** Learned starting from strategy 4; 2000 games self-play
6	**Hand-set 3.** Hand-set weights.
7	**Learn 3.** Learned starting from strategy 6; 2000 games self-play
8	**Learn 4.** Learned starting from strategy 6; 1240 games, fixed opponents
9	**DFS 1.** 2-ply DFS using strategy 1 evaluation function
10	**Beam Search 1.** 5-ply beam search using strategy 1 evaluation function
11	**DFS 2.** 2-ply DFS using strategy 2 evaluation function
12	**Beam Search 2.** 5-ply beam search using strategy 2 evaluation function
13	**DFS 3.** 2-ply DFS using strategy 3 evaluation function
14	**Beam Search 3.** 5-ply beam search using strategy 3 evaluation function
15	**DFS 4.** 2-ply DFS using strategy 7 evaluation function
16	**Beam Search 4.** 5-ply beam search using strategy 4 evaluation function

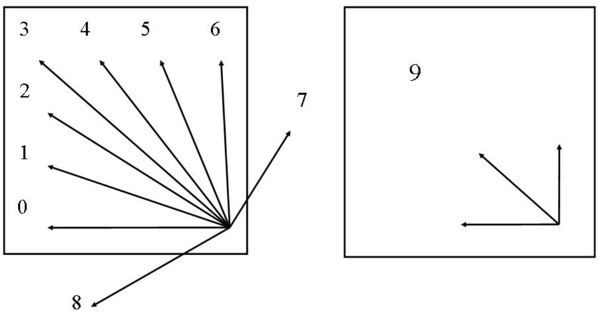

Figure 2: A visualization of the attack angle parameter.

gain a better understanding of the effects of the attack angle parameter on the outcome of the game we ran several linear regressions which are shown in Table 2. As expected, players that have more "attack weight" placed on them tend to have lower scores on average. The most potent attacks come from the player to the right of a given player because this player can employ advancing pawns directly in the attack. Attacks from the diagonal players are largely ineffective (very low R^2), probably because of the spatial separation. On the other hand, players benefit noticeably when *other* players direct their attention to the diagonal player (and to a lesser degree, the player on the right). Each player relies on other players to defeat opponents that it does not have strong position to attack, but this may not be in the best interest of the other players.

Based on our data for the exploratory strategy space we have identified two pure-strategy Nash equilibria of the empirical game, and two other interesting profiles. These are shown in Table 3. The profile [15,15,16,16] has a beneficial deviation to the equilibrium profile [15,16,16,16]. The beneficial deviation for the profile [16,16,16,16] is to the equilibrium profile [16,16,16,15]. There are no other candidate equilibria in the data set that have an epsilon bound less than 0.05 and fewer than 44 unexplored deviations. For the attack angle strategy space there are no pure-strategy Nash equilibria of the empirical game. The 10 most stable profiles are shown in Table 4.

Additional statistics about each of the individual strategies are given in Tables 5 and 6. These include both measures of average performance across the entire data set and the likelihood that each strategy appears in stable profiles. The strategies are listed in order of increasing average scores. The average performance metrics and profile stability metrics are remarkably strongly correlated, a point we return to after some remarks about each strategy space.

By any measure, strategies 15 and 16 are clearly the strongest in the exploratory strategy space. They are the only ones to appear in the pure-strategy Nash equilibria, and have much higher average scores than all other strategies. These are the two strategies that incorporate both a learned evaluation function and online search. The beam search version (16) is more common in the stable profiles, but 15 is a best response in some cases. The strategies with learned evaluation functions did very well overall; the three versions without online search finished next in the average score ordering, above even the hand-set evaluation functions with online search. Among these three, the ones with the most training data performed better. The robust performance of the learned evaluation functions in this setting shows that they were able to learn strategies that generalized well to novel situations.

Online search also seems to have improved the performance of the strategies. Each of the evaluation functions used in strategies 1, 2, 3, and 7 had two search versions: one DFS (9,11,13,15) and one beam (10,12,14,15). In all cases except evaluation function 2, the search versions appear to be stronger than the non-search version. All versions using evaluation function 2 were very weak. There is not a clear winner between the DFS and beam search variants. This may depend on the strength of the evaluation function, as it seems that the weaker evaluation functions (1,3) fared better with DFS than the stronger evaluation function (7).

In comparison with the exploratory strategy space, the attack angle space appears to be much more strategically interesting. None of the strategies is clearly dominant, and all appear in at least one relatively stable profile. Additionally, none of the pure strategy profiles is a Nash equilibrium. One noticeable trend in the data is that strategies that distribute weight across more opponents seem

Table 2: Data showing the effects of attack angle settings on average score. Total weight is the sum of the weight placed on the relevant player from all three opponents.

	R^2	Coefficient
Total weight on player	0.36	–0.14
Weight from right player	0.25	–0.20
Weight from left player	0.14	–0.07
Weight from diagonal player	0.03	–0.15
Total weight on right player	0.06	0.05
Total weight on left player	0.002	0.26
Total weight on diagonal player	0.17	0.09

Table 3: Most stable profiles for the exploratory strategy space

ϵ	Unexplored Deviations	Profile	Scores
0	0	[15,16,16,16]	[0.15,0.25,0.30,0.30]
0	0	[16,16,16,15]	[0.23,0.15,0.37,0.25]
0.05	0	[15,15,16,16]	[0.33,0.20,0.23,0.23]
0.02	1	[16,16,16,16]	[0.13,0.30,0.33,0.23]

Table 4: Most stable profiles for attack angle strategy space.

ϵ	Profile	Scores
0.04	[7,8,8,2]	[0.22,0.23,0.21,0.33]
0.05	[8,7,4,7]	[0.24,0.38,0.27,0.11]
0.05	[7,1,8,8]	[0.16,0.29,0.18,0.37]
0.06	[2,5,8,7]	[0.09,0.26,0.52,0.12]
0.06	[9,7,4,7]	[0.25,0.46,0.17,0.13]
0.06	[9,9,9,7]	[0.17,0.37,0.23,0.24]
0.06	[9,7,5,8]	[0.40,0.12,0.32,0.16]
0.06	[9,9,5,3]	[0.44,0.25,0.15,0.17]
0.06	[4,0,7,7]	[0.26,0.24,0.12,0.38]
0.06	[4,7,8,6]	[0.12,0.42,0.15,0.31]

Table 5: Statistics about the performance of individual strategies in the exploratory strategy space. Score is the average score over all profiles explored. Dev Benefit is the average benefit for deviating from the strategy over all deviations explored. % Positive Dev is the percentage of deviations from this strategy that are beneficial. The top X% are the percentage of times the given strategy is played in the most stable (lowest ϵ-bound) profiles, considering only profiles with at least 20 out of 64 possible deviations explored.

Strategy	Score	Dev Benefit	% Positive Dev	top 5%	top 1%	top 0.1%
3	0.049	0.160	0.779	0.005	0	0
0	0.051	0.183	0.811	0.003	0	0
11	0.054	0.146	0.757	0.012	0.008	0
12	0.064	0.143	0.720	0.012	0	0
2	0.089	0.148	0.722	0.017	0.008	0
1	0.105	0.131	0.675	0.017	0.008	0
14	0.110	0.078	0.586	0.008	0	0
10	0.177	0.034	0.476	0.017	0.008	0
6	0.199	0.019	0.474	0.025	0.025	0
9	0.213	0.019	0.479	0.025	0.017	0
4	0.221	0.009	0.442	0.034	0.017	0
13	0.255	-0.011	0.396	0.035	0.042	0
8	0.280	-0.041	0.342	0.032	0.025	0
7	0.353	-0.110	0.279	0.037	0.033	0
5	0.383	-0.155	0.210	0.083	0.050	0
15	0.480	-0.266	0.074	0.277	0.225	0.167
16	0.481	-0.261	0.083	0.360	0.533	0.833

Table 6: Statistics about the performance of individual strategies in the attack angle strategy space.

Strategy	Score	Dev Benefit	% Positive Dev	top 5%	top 1%	top 0.1%
0	0.214	0.04	0.663	0.045	0.022	0.025
6	0.222	0.031	0.627	0.047	0.025	0.025
3	0.238	0.013	0.549	0.082	0.055	0.025
1	0.245	0.005	0.519	0.083	0.052	0.025
5	0.246	0.004	0.512	0.098	0.125	0.075
2	0.248	0.003	0.503	0.090	0.068	0.050
4	0.255	-0.006	0.47	0.118	0.120	0.100
7	0.263	-0.015	0.432	0.132	0.152	0.300
8	0.279	-0.032	0.364	0.143	0.162	0.200
9	0.289	-0.043	0.321	0.162	0.218	0.175

to do better. The three lowest strategies are the ones that “attack” a single opponent, while the highest is the one that weights all three opponents equally.

As noted above, there is a striking correlation between the aggregate performance measures and the profile stability metrics. Tables 7 and 8 give the actual correlations for each strategy space. This correlation is interesting because it is consistent with the hypothesis that there is some form of strategic independence in the underlying game. There are many possible ways to exploit this sort of structure. In addition the local search techniques discussed previously, this could also function as a reasonable search heuristic for guiding the choice of which profiles to sample next in the best-first search procedure for finding equilibria.

7. DISCUSSION

We have presented a preliminary analysis of the interesting and challenging game chaturanga using methods from the emerging field of empirical game theory. The focus of the discussion has been on the notion of strategic independence and the possible application of this idea to the analysis of large games. Our preliminary experiments with chaturanga suggest that strategic independence exists and can be exploited, at least in this game. We will continue to seek opportunities to improve these and other methods for analyzing empirical games.

8. ACKNOWLEDGMENTS

We would like to thank Matt Abrams and the rest of Cougaar Software for developing the infrastructure for playing chaturanga games. We would also like to thank Ryan Porter, Michael Kearns, and Patrick Lincoln for their help in developing the chaturanga strategy space and interesting discussions about the game. This project was supported by the DARPA Real strategic reasoning program.

9. REFERENCES

[1] J. Baxter, A. Tridgell, and L. Weaver. Learning to play chess using temporal-differences. *Machine Learning*, 40(3):243–263, 2000.

[2] M. Glickman. Chess rating systems. *American Chess Journal*, 3:59–102, 1995.

[3] J. Harsanyi. Games with incomplete information played by Bayesian players, Parts I, II, and III. *Management Science*, 14:159–182, 320–334, 486–502, 1967–1968.

[4] M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 253–260, 2001.

[5] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45(1):181–221, 2003.

[6] C. A. Luckhardt and K. B. Irani. An algorithm for the solution of n-person games. In *Fifth National Conference on Artificial Intelligence*, pages 158–162, 1986.

[7] M. Lustrek, M. Gams, and I. Bratko. Why minimax works: An alternative explanation. In *Nineteenth International Joint Conference on Artificial Intelligence*, pages 212–217, 2005.

[8] A. Osepayshvili, M. P. Wellman, D. M. Reeves, and J. K. M. Mason. Self-confirming price prediction for bidding in simultaneous ascending auctions. In *Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 441–449, 2005.

[9] J. Park and M. E. J. Newman. A network-based ranking system for American college football. *Journal of Statistical Mechanics*, 2005.

[10] S. Phelps, M. Marcinkiewicz, and S. Parsons. A novel method for automatic strategy acquisition in n-player non-zero-sum games. In *Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, 2006.

[11] D. M. Reeves, M. P. Wellman, J. K. MacKie-Mason, and A. Osepayshvili. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39:67–85, 2005.

[12] C. D. Rosin and R. K. Belew. Methods for competitive co-evolution: Finding opponents worth beating. In *Sixth International Conference on Genetic Algorithms*, pages 373–380, 1995.

[13] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3:211–229, 1959.

[14] R. T. Stefani. Survey of the major world sports rating systems. *Journal of Applied Statistics*, 24(6):635–646, 1997.

[15] N. Sturtevant. *Multi-Player Games: Algorithms and Approaches*. PhD thesis, UCLA, 2003.

[16] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press/Bradford Books, 1998.

[17] Y. Vorobeychik, C. Kiekintveld, and M. P. Wellman. Empirical mechanism design: Methods, with application to a supply-chain scenario. In *Seventh ACM Conference on Electronic Commerce*, 2006. To appear.

[18] W. E. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*, 2002.

[19] M. P. Wellman, J. Estelle, S. Singh, Y. Vorobeychik,

Table 7: Correlations for exploratory strategy space.

Strategy	Score	Dev Benefit	% Positive Dev	top 5%	top 1%
Dev Benefit	–0.99				
% Positive Dev	–0.99	0.99			
top 5%	0.80	–0.82	–0.75		
top 1%	0.71	–0.73	–0.66	0.96	
top 0.1%	0.57	–0.60	–0.53	0.88	0.98

Table 8: Correlations attack angle strategy space.

	Score	Dev Benefit	% Positive Dev	top 5%	top 1%
Dev Benefit	–1.00				
% Positive Dev	–1.00	1.00			
top 5%	0.98	–0.98	–0.99		
top 1%	0.94	–0.94	–0.94	0.97	
top 0.1%	0.74	–0.75	–0.75	0.80	0.80

C. Kiekintveld, and V. Soni. Strategic interactions in a supply chain game. *Computational Intelligence*, 21:1–26, 2005.

[20] M. P. Wellman, D. M. Reeves, K. M. Lochner, and R. Suri. Searching for Walverine 2005. In *IJCAI-05 Workshop on Trading Agent Design and Analysis*, 2005.