Optimal Risk in Multiagent Blind Tournaments, with application to Yahtzee

Theodore J. Perkins
Ottawa Hospital Research Institute
Ottawa, Ontario, Canada
tperkins@ohri.ca

ABSTRACT

In multiagent blind tournaments, many agents compete at an individual game, unaware of the performance of the other agents. When all agents have completed their games, the agent with the best performance—for example, the highest score, or greatest distance, or fastest time—wins the tournament. In stochastic games, an obvious and time honoured strategy is to maximize expected performance. In tournaments with many agents, however, the top scores may be far above the expected score. As a result, maximizing expected score is not the same as maximizing the chance of winning the tournament. Rather, a "riskier" strategy, which increases the chance of obtaining a top score while possibly sacrificing some expected score, may offer a better chance of winning. In this paper, we study how an agent should optimally adapt its strategy based on the size of the tournament in which it is competing. Our solution involves first approximating the agent's pool of opponents as a collection of known or estimated strategies. Second, score distributions are computed for those strategies, and the distributions are convolved to obtain a distribution for the maximum score of the agent's opponents. Finally, a strategy that maximizes the chance of exceeding the opponents' scores is computed. As a demonstration, we study optimal tournament-size adaptation in online Yahtzee tournaments involving as few as one and as many as ten thousand opponents. We find that strategies change dramatically over that range of tournament sizes, and that in large tournaments, an agent adopting the optimally risky strategy can nearly double its chance of winning.

KEYWORDS

multiagent; tournaments; adaptation; optimality; risk; Yahtzee

1 INTRODUCTION

In many games, agents compete directly against each other and make many tactical and strategic decisions based on their knowledge and observations of their opponents [13, 14, 17]. In other games, however, the agents perform individually, and winning is based on which agent performs the best. Examples include some types of auctions [11], certain game shows, applying for a job or an exclusive fellowship or a grant, and the original motivation for the present work, massively multiagent online tournaments. When many individuals are competing, the bar for success—say, the winning game score in a tournament—is naturally higher. Thus, although each agent is playing individually, an intelligent agent must adapt its strategy based on, at the very least, the size of tournament in which it is competing.

Our specific motivation for the present study was the online Yahtzee with Buddies game. For those not familiar with the game of Yahtzee, we describe it in detail below. Briefly, it is a turn-based dice game in which agents get points for different combinations of dice. When humans play Yahtzee with real dice, they can see each others' rolls and scores, and may make strategic decisions based on that information. The Yahtzee with Buddies app has a similar one-on-one mode of play, but it also offers tournament play. In tournament play, each agent plays a game alone, and when all agents have completed their games, the agent with the highest score wins. The app offers tournaments with as few as two other agents and daily tournaments with an unlimited number of agents (typically around ten thousand). In playing such tournaments, it rapidly becomes obvious that the winning strategy for a small tournament differs from the winning strategy for a large tournament. Large tournaments demand risky play, because the score needed to win a large tournament can be exceedingly high. Thus, we became interested in the general phenomenon of risk-taking in tournaments, and how optimal play must adapt to the size of the tournament.

Of course, there has been extensive study of risk, risk-reward trade-offs, and related concepts in fields such as game theory, decision theory, and economics [5, 9]. In investment theory, portfolio management is often concerned with the problem of growing investments while minimizing risk. For any individual investment, however, risk and reward are often viewed as going together—the only way of achieving a high potential (or expected) payoff, is to accept the possibility of substantial loss as well. This is related to the notion of risk we observe in tournament games. Maximizing the chance of winning a large tournament necessarily means accepting the possibility of one's game coming out very poorly. However, we are not aware of prior work specifically on the question of optimal risk as a function of the number of competing agents, and certainly not in the context of Yahtzee.

Prior work on Yahtzee has mostly focused on the problem of computing strategies that maximize expected score. Optimal strategies were computed independently by Verhoeff [19], Woodward [20], and Glenn [8]. Cremers considered the problem of how to maximize the chance of beating any given score threshold S [6]. Implicitly, this creates a similar "optimal risk" problem to the one we study, in the sense that aiming for high S requires taking greater risks. Our parameterization in terms of the number of opponent players, however, results in a different problem. Firstly, the necessary "winning threshold" is not obvious as a function of the number of opponents. Further, as we will show in detail below below, winning score distributions are complex and not well approximated by a single "winning threshold." To our knowledge, only Pawlewicz [15] has seriously studied a multi-agent version of Yahtzee, with a focus on beating either one or a collection of opponents. They explored heuristic strategies for winning in multiplayer games. In contrast, we focus on computing optimal strategies for winning blind Yahtzee tournaments, and study how the strategy changes as a function of the number of competitors.

In this paper, we propose a general approach to defining optimal agent behaviour in multiagent blind tournaments. To achieve maximum generality, we sought to minimize the assumptions made by our approach. In its simplest form, we require only: (1) that the rules of the game are known; (2) that playing the game results in a discrete-valued score; (3) that the single-agent version of the game can be solved optimally for any terminal objective function that depends on score; and (4) that the number of opposing agents in the tournament is known. Our solution concept also allows the agent to have models of the opposing agents' strategies (as in for example [2, 4, 7]). The organization of this paper is a follows. First, we define multiagent blind tournament games. Second, we lay out our solution strategy. Third, we define the rules of our example domain, Yahtzee. Fourth, we describe our dynamic programming approach to compute optimal Yahtzee strategies, which is non-trivial as Yahtzee has a challengingly-large state space. Finally, we present empirical data from the online game alongside optimal strategies for different sizes of Yahtzee tournaments. We emphasize how optimal strategies adapt to different tournament sizes, and in particular, how higher risk play is optimal in larger tournaments.

2 DEFINITIONS AND PROBLEM STATEMENT

We define a single-agent *game* to be a tuple $\mathcal{G} = (G, S, A, \mathcal{A}, P, g_0, T)$, where:

- $G = \{g_1, \ldots, g_l\}$ is a set of possible game states,
- $S : G \rightarrow \Re$ defines the agents score in every game state,
- $A = \{a_1, \ldots, a_n\}$ is a set of possible agent actions,
- $\mathcal{A}(g) : G \mapsto A$ specifies the allowed agent actions in every game state,
- $P : G \times A \times G \mapsto [0, 1]$ is the transition function, where $P(g, a, g')$ gives the probability that game state g' follows when an agent takes action $a \in \mathcal{A}(g)$ from game state g,
- $g_0 \in G$ is the initial game state, and
- $T \subset G$ is a nonempty set of terminal or end game states. From these states, no further actions are allowed and no game state transitions are possible.

A game is similar to a Markov Decision Process (MDP) [16], but it lacks an explicit reward or cost function. The score component of the game state can be viewed as one notion of reward. However, we will not be assuming that the goal is to maximize the expected score. Indeed, the idea central to this paper is that maximizing expected score can be quite different from maximizing the chance of winning a tournament.

A *policy* π maps every game state $g \in G$ to either a single action allowed in that game state, or to a distribution over such actions. Whether the policy is deterministic or stochastic, we let $\pi(g, a)$ denote the probability of taking action a in game state g.

A *tournament* describes a situation in which one or more agents play *statistically independent* instances of the same game $\mathcal{G}$. We assume for simplicity that every agent that competes in the tournament completes its game, and thus obtains a final score. (However, it is straightfoward to extend our framework to allow for some agents not completing their games.) If there is a single agent with a final score higher than all other agents' final scores, then that agent is declared the winner of the tournament. Otherwise, there is no winner.

In general, when an agent is playing in a tournament, the optimal strategy may depend on who else is playing in the tournament and what their strategies are. This information may not be available, and researchers in game theory have proposed solution concepts, such as Nash equilibria, as both descriptive but also proscriptive theories [14]. However, sometimes an agent does know who the opponents are and what their strategies are—for instance through some approximation or past experience—and this can be used to adapt one's strategy (e.g. [2, 4, 7]). This is the approach we take, with the main novelty being our emphasis on how tournament size influences strategy. Formally, the problem we seek to solve is: *Find a policy that maximizes the chance of winning a tournament of game $\mathcal{G}$ against N other agents who follow known policies $\pi_1, \ldots, \pi_N$.* In the next section, we describe how this can be done in relatively straightfoward fashion, at least conceptually. We then describe our application of this solution strategy to tournament Yahtzee, where computing with the large state spaces involved is the primary challenge.

3 SOLUTION APPROACH

The first observation in our solution is that the combination of a game $\mathcal{G} = (G, S, A, \mathcal{A}, P, g_0, T)$ and a policy π defines a Markov process $\mathcal{M} = (G', P', g'_0, T')$ in the following obvious way. The state set is the same as the game states $G' = G$. The transition probabilities are $P'(g, g') = \sum_{a \in \mathcal{A}(g)} \pi(g, a) P(g, a, g')$. The initial state is the same, $g'_0 = g_0$. Finally, the terminal states are the same, $T' = T$.

Second, as for any Markov process with an initial state and terminal states, there is a (sub)probability distribution describing the chance that the process eventually reaches each terminal state [16]. More formally, if g_t is a random variable denoting the state of the process at time t, then the distribution of interest is $\mu_\pi(g) = \lim_{t \to \infty} P(g_t = g)$ for each $g \in T$. We use the subscript π on μ to make explicit the dependence of the terminal distribution on the policy π. For a Markov process in general, this is a subprobability distribution. We made the simplifying assumption above, however, that all agents complete their games. More formally, this means that π causes the game to reach a terminal state for certain, or at least with probability one. In most games, no plausible policy leads to an endless game, and in some games, such as Yahtzee, every policy causes the game to end. The equations below, however, can be generalized to account for policies causing unending games.

The third observation is that for an opponent agent's policy π, the full distribution over terminal states of the Markov process is not important. We really only need to know the terminal distribution over scores, as it is the scores that determine the winner of the tournament. Thus, we abuse our notation for terminal distribution somewhat to also define a terminal distribution over scores: $\mu_\pi(s) = \sum_{g : S(g) = s} \mu_\pi(g)$. Note that scores are real-valued, but that the set of states G' (or G) is finite. Therefore, the final score distribution is a discrete distribution.

Now, recall that our problem statement involves one agent in a tournament trying to beat N other agents, which play strategies $\pi_1, \ldots, \pi_N$. Let $o_1, \ldots, o_N$ be random variables describing the final scores of those agents—their "outcomes". Let $M_j = \max_{i=1}^{j} o_i$ be the maximum score obtained by the first $j \leq N$ agents. To beat all the other agents, our agent must score strictly greater than M_N. What is the distribution of M_N? It can readily be obtained by convolving the individual agents' score distributions recursively as follows.

$$Pr(M_1 = s) = Pr(o_1 = s) = \mu_{\pi_1}(s) \ .$$

For $j > 1$,

$$\begin{aligned}
& Pr(M_j = s) \\
= \quad & Pr\left(\max_{i=1\ldots j} o_i = s\right) \\
= \quad & Pr(\max(o_j, M_{j-1}) = s) \\
= \quad & Pr((o_j = s \text{ and } M_{j-1} \leq s) \text{ or } (o_j < s \text{ and } M_{j-1} = s)) \\
= \quad & \mu_{\pi_j}(s)\left(\sum_{s' \leq s} Pr(M_{j-1} = s')\right) + \left(\sum_{s' < s} \mu_{\pi_j}(s')\right) Pr(M_{j-1} = s)
\end{aligned}$$

Having used these equations to compute $Pr(M_N = s)$, we can then write the probability of our agent winning with a score of s as $W(s) = Pr(M_N < s) = \sum_{s' < s} Pr(M_N = s')$.

Finally, we can state our approach to finding an optimal strategy for playing the tournament. We construct a Markov decision process $\mathcal{MDP} = (G'', A'', \mathcal{A}'', P'', g_0'', T'', R)$ where:

- $G'' = G$ is the set of possible (game) states,
- $A'' = A$ is the set of possible actions,
- $\mathcal{A}'' = \mathcal{A}$ tells which actions are possible from which states,
- $P'' = P : G'' \times A'' \times G'' \longrightarrow [0, 1]$ is the transition function,
- $g_0'' = g_0$ is the initial state,
- $T'' = T$ are the terminal states,
- R is the reward function, defined as follows:

$$R(g, a, g') = \begin{cases} W(S(g')) & \text{if } g \notin T'' \text{ and } g' \in T'' \\ 0 & \text{otherwise} \end{cases}$$

In words, the MDP follows the same dynamics as a single game, but upon completion of the game, the final reward is equal to that chance that the final score obtained will win the tournament. By construction, the policy that maximizes this expected reward maximizes the chance of winning the tournament. Incidentally, this construction also implies that a deterministic strategy is sufficient for maximizing the chance of winning the tournament (as MDPs are in general optimally solved by deterministic policies [16])—as opposed to a stochastic policy, or history-based policy, or some other more complicated construct. We next turn to Yahtzee, where we provide some details of the game, and describe how we applied the above solution concept to computing strategies for Yahtzee tournaments.

Table 1: Scoring rules for each of 13 scoring boxes in Yahtzee

Box	Scoring rule	Example
Ones	1 × no. of ones	⚀⚀⚁⚂⚄ → 2 points
Twos	2 × no. of twos	⚀⚁⚁⚁⚅ → 6 points
Threes	3 × no. of threes	⚀⚁⚃⚃⚄ → 0 points
Fours	4 × no. of fours	⚃⚃⚃⚃⚄ → 16 points
Fives	5 × no. of fives	⚁⚂⚃⚄⚅ → 5 points
Sixes	6 × no. of sixes	⚅⚅⚅⚅⚅ → 30 points
Three of a kind	Sum of dice if ≥ 3 match	⚁⚃⚃⚃⚄ → 19 points
Four of a kind	Sum of dice if ≥ 4 match	⚁⚃⚃⚃⚄ → 0 points
Full house	25 if 2 match & 3 others match	⚁⚁⚁⚄⚄ → 25 points
Small straight	30 if four dice in series	⚀⚂⚃⚄⚅ → 30 points
Large straight	40 if five dice in series	⚀⚁⚂⚃⚄ → 40 points
Yahtzee	50 if all five dice match	⚁⚁⚁⚁⚁ → 50 points
Chance	Sum of dice	⚁⚂⚃⚄⚅ → 20 points

4 YAHTZEE RULES

A single-agent Yahtzee game comprises 13 turns, and each turn involves at least one and up to three rolls of dice. At the start of each turn, the agent rolls five dice. The agent can then "assign" that roll to one of 13 distinct scoring boxes (unless a box has already been used). The points obtained depend on the dice faces showing as well as the box, as summarized in Table 1. Alternatively, the agent can re-roll some or all of the dice. Then, there is another decision to either assign the dice roll to a scoring box, or to re-roll some or all of the dice yet again. If three rolls have been made, then the dice roll must be assigned to one of the unused scoring boxes.

The total score for the game depends on the sum of the individual box scores, but with two additional rules. First, if the sum of the scores in the boxes Ones through Sixes it at least 63, then 35 bonus points are awarded. Second, if the agent has already scored a Yahtzee (worth 50 points), and if the agent rolls any subsequent Yahtzees and assigns them to one of the scoring boxes, then an additional 50 points is awarded for each such Yahtzee. Such points do not count towards the 63 bonus. If the agent "cancels" or "scratches" the Yahtzee box, by assigning to it a roll in which all five dice do not match, then any subsequent Yahtzees they may roll do not earn the 50 additional points. Note that it is allowed to assign any roll to any unused box, even if no points are obtained. In the official Yahtzee rules, additional Yahtzees are worth 100 points, not 50. However, we use 50 to be consistent with the online Yahtzee with Buddies game. With these rules, the maximum obtainable score is 880, which comes from rolling a Yahtzee every turn, including a Yahtzee of ones for the Ones box, a Yahtzee of twos for the Twos box, and so on. The minimum score is five, which is obtained by getting a zero in every box except chance, where one places a roll of five ones (after scratching the Yahtzee box).

5 STATE REPRESENTATION AND DYNAMIC PROGRAMMING

The full state of a Yahtzee game is too large to be conveniently represented explicitly. To see this, observe that the Ones box can be in one of seven states: unused, or used with a score of between zero and five. The Twos box can also be in one of seven states: unused, or used with a score of 0, 2, 4, 6, 8 or 10. Similarly with the Threes through Sixes boxes. The Three of a Kind box can be have 26 states: unused, or used with a score of 0, or $7, 8, 9, \ldots, 30$. And so on. Leaving the details to the reader, one can see that in total there are $7^6 \cdot 26 \cdot 27 \cdot 3^4 \cdot 28 = 187,313,208,264$ possible joint states for the scoring boxes. One must also account for the number of additional Yahtzees, between zero and twelve, beyond the first Yahtzee. Although we obviously cannot have as many as 12 additional Yahtzees in every possible joint scoring box state—for example, if only 3 turns have been played—we can naively bound the total number of states as $13 \times 187,313,208,264 = 2,435,071,707,432$. Finally, there is the matter of the dice. Accounting for symmetries, five six-sided dice can show 252 distinct combinations. Given that the dice may be unrolled, at the start of a turn, or show 252 combinations after one, two or three rolls during a turn, the dice add $1 + 3 \times 252 = 757$ possibilities. This leads us to 1.843 quadrillion possible full Yahtzee game states. If we required an 8-byte floating point number to store the value of each game state—representing the chance of winning the game, for example—this would equate to approximately 16 petabytes of storage. Such an amount is available on higher-end computing clusters or data centers, as is the computing power necessary to determine the values of those game states. However, we adopt a more parsimonious representation.

Our game state representation can be divided into four main components: 1) UsedState: indicating which scoring boxes have been used; 2) TopScore: the sum of scores obtained so far in the Ones through Sixes boxes, not counting any bonus; 3) BottomScore: the sum of scores obtained so far from the other boxes, Three-of-a-kind through Chance, including scores for any additional Yahtzees beyond the first; 4) RollState: the state of the dice. The first component, UsedState, comprises a binary indicator for each scoring box except Yahtzee, indicating whether or not the box has been used. For the Yahtzee box, we have a trinary value, indicating: unused, used but no Yahtzee scored, and used with a Yahtzee scored. The distinction between the two notions of the Yahtzee box being used are necessary because any future Yahtzees only accrue the bonus if a Yahtzee was first scored in the Yahtzee box. There are $2^{12} \cdot 3 = 12288$ distinct UsedStates. TopScore, so called because the Ones through Sixes boxes are in the top half of a traditional paper-based Yahtzee scoring sheet, is an integer ranging from zero to 105, the latter achieved with five dice of the correct type being scored in every box. BottomScore is an integer ranging from zero to 740, the latter achieved with Yahtzees of sixes assigned to the Three-of-a-kind, Four-of-a-kind and Chance boxes, any type of Yahtzee assigned to the Full House, Small Straight, Large Straight and Yahtzee boxes, and Yahtzees in all of the Ones through Sixes boxes. As described above, the RollState is an integer ranging from zero to 756, indicating whether the dice have not been rolled yet (one possibility), or rolled once, twice or thrice.

In essence, then, we use four dimensional tables, indexed by UsedState (0 to 12287), TopScore (0 to 105), BottomScore (0 to 740) and RollState (0 to 756), to represent the values or optimal actions associated to different game states. However, the maximum possible TopScore and BottomScore values are not achievable in all UsedStates. So, we create a set of 12288 three dimensional tables, one for each possible UsedState, and with TopScore and BottomScore ranging over only their possible values depending on the scoring boxes used in each UsedState. Two different UsedStates correspond to a completed game, one where a Yahtzee has been scored and one where Yahtzee has been canceled. For these two UsedStates, the RollState only has one possible value—“unrolled" or “game over". Our collection of tables has $100,706,855,944 \approx 100$ billion entries.

The actions available to an agent in any of these game states is based on the Yahtzee rules, as described in the previous section, and the state transition probabilities follow what one would expect. Rolling the dice results in no change in UsedState, TopScore or BottomScore, but results stochastically in a new RollState. That RollState will indicate that one additional roll has been taken. We computed by straightforward enumeration the probabilities of different combinations of dice faces resulting based on the original dice faces and which dice were being (re)rolled. Assigning a dice roll to a score box results in a change in the UsedState (indicating the box that was used), and may result in a change to TopScore, BottomScore or both (in the case that the agent rolls a bonus Yahtzee and assigns it to one of the Ones through Sixes boxes). The RollState resets to zero, indicating either an unrolled dice state at the start of the next turn or that the game is over.

We implemented standard dynamic programming techniques to calculate optimal policies, state values, and state probabilities [16], for different possible reward functions, as will be described below. More specifically, backwards dynamic programming was used to compute state values and optimal actions for every game state, starting with the terminal states. Then values of states where there is one turn left were computed based on the terminal states values, and so on. We scheduled the computations as one process per UsedState on the Compute Canada cluster Frontenac, at the Centre for Advanced Computing. Because the values of game states with different UsedStates but the same number of turns cannot depend on each other, we scheduled the computations of all such UsedStates to proceed in parallel. Only when all computations for a particular number of turns are completed do we proceed to computations for the previous turn. Once an optimal policy is computed, we then perform a forward dynamic program to compute the probabilities of different game states [16], culminating in a policy-dependent distribution over terminal states, and hence a terminal score distribution. Allowing for some manual verification and restarting of dropped processes, the entirety of one such computation took approximately 1-2 weeks for a single choice of reward function.

6 EMPIRICAL WINNING-SCORE DISTRIBUTIONS

To get a sense of how high scores are needed to win in multiagent-blind Yahtzee tournaments of different sizes, we manually played tournament games online, via the Yahtzee with Buddies gaming app. This app allows agents to compete in tournaments of different sizes. Not all sizes of tournaments are available at all times. But over time, we were able to accumulate 50 played games in 15-agent tournaments, 25-agent tournaments and 50-agent tournaments. In these games, we played a strategy of indefinite optimality—one that we had developed through prior manual play, and which we believe is “good" but not the best possible. As evidence that our strategy is good, we won 5 of 50 15-agent tournaments, 3 of 50 25-agents tournaments, and 1 of 50 50-agent tournaments. All these empirical win rates are at or above expected level if all agents were acting optimally (or following any identical strategy). At the end of these tournaments, the scores of the top-3 scoring agents are revealed, including of course the winner, as well as ones’ own rank among the agents. The app also offers two daily tournaments that allow an unlimited number of agents. We also played 25 of each of these tournaments. For these, we attempted to get the lowest score possible. The reason for this is that, from the outcome of the tournament, we could see the scores of the top-scoring agents, but we could not see how many agents were competing. By scoring as low as possible, often as little as 6-8 points in a whole game, we could use our rank in the tournament standings as an approximation of the number of agents that had competed. We found that one of the two daily tournaments consistently had 5,000-6,000 agents, while the other had 10,000-12,000. (The difference in the tournaments is in the rewards for winning or placing high, which presumably effects their desirability to players.) Figure 1 shows the empirical spread of winning scores in tournaments of these different sizes. As one would expect, winning scores generally increase as a function of the number of agents in the tournament. The variability in winning score decreases with tournament size. This is to be expected from first principles, for example on the basis that the maximum of N samples from a bounded distribution has variance that decreases in N [1].

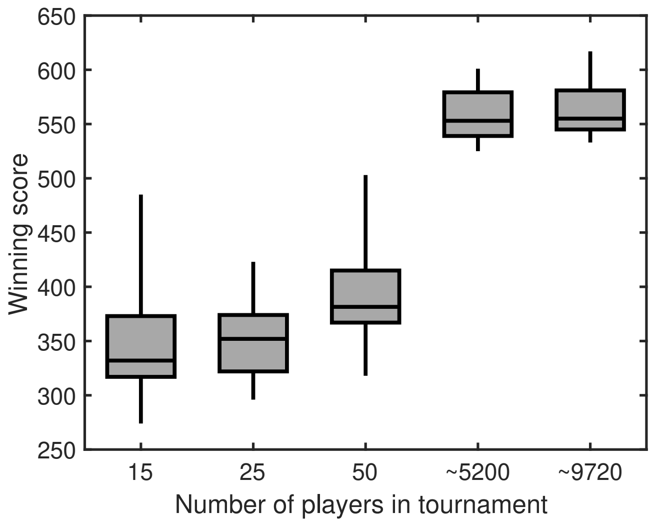

Figure 1: Distributions of the winning (top) score in multiagent-blind Yahtzee tournaments of different sizes, based on manual play in the “New Yahtzee with Buddies" app. For each tournament size, the minimum, 25th percentile, median, 75th percentile and maximum winning scores across the sampled tournaments are shown.

7 PLAY THAT MAXIMIZES EXPECTED SCORE

Next, we computed a policy that maximies the expected score. This has been done several times before [8, 20], though not always with the exact same rules, and to our knowledge, no one has previously exactly computed the distribution of scores for that optimal policy. We did this by dynamic programming as described above, choosing as terminal reward the game score, and with no (i.e. zero) intermediate rewards. The maximum expected score achievable is approximately 243.08. The black curve in Figure 2A shows the probability distribution over final scores achieved by that policy. It is interestingly complex, with many distinct components. The components are mainly due to whether or not a Yahtzee is scored, whether or not a large straight is scored, and so on. In contrast, categories like Ones through Sixes, Three-of-a-kind, Four-of-a-kind, and Chance, where one can score a wide range of values, tend to contribute to the smoother aspects of the distribution.

Based on the score distribution of a single score-maximizing agent, we calculated the distribution for the maximum score out of 10, 100, 1000, and 10,000 agents, as described in Section 3 (Figure 2A). The most obvious effect of increasing the number of competing agents is that the distributions shift toward higher values. In a 10-agent tournament, the median winning score is just under 300, whereas it is approximately 500 when 10,000 agents are competing.

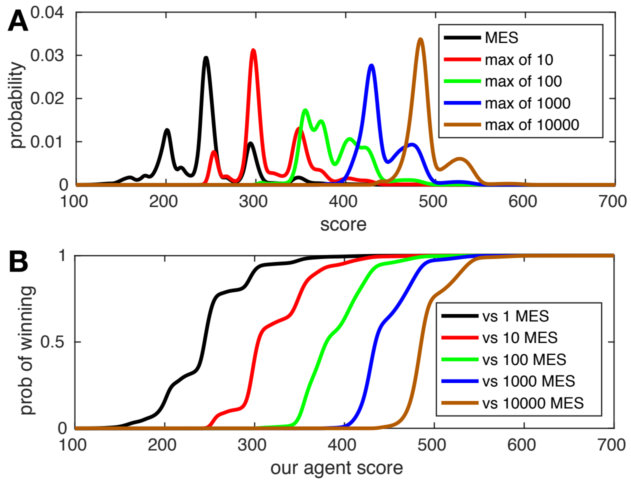

Figure 2: Score distributions for agents maximizing expected score, and the chances of winning tournaments against them. (A) The probabilities of different scores when a single agent plays the strategy that maximizes expected score (MES), and the probabilities of different maximum scores when 10, 100, 1000 or 10000 MES agents play statistically independent games. (B) The chance of an individual agent winning a tournament against, 1, 10, 100, 1000, or 10000 opponent MES agents.

Note that even in a 10,000 agent tournament, the distribution of the highest score is still substantially bimodal. At these high scores, agents are getting multiple Yahtzees, and the two modes in the distribution are likely due to getting one more or one fewer Yahtzee.

Comparing Figures 1 and 2A, we note some discrepancy in estimated winning scores for tournaments of approximately 10,000 participants. Empirically, in the Yahtzee with Buddies tournaments, the median winning score exceeds 550, more than 50 points higher than our calculated median winning score for 10,000 expected-score maximizing agents. There are two obvious possible reasons for this. One is that participants in the online game are not playing to maximize expected score, but are, intelligently, playing a more risky strategy in order to increase the chance of winning. However, another likely explanation is that the online game has a sort of currency, "extra dice," which one can buy with real money or earn in other ways on the app. One can spend a limited number of these during the game to obtain extra rolls, which allows one to obtain better scores. Our model does not account for this aspect of the online game, which is absent in the Yahtzee box sets one can buy. Regardless, in either the app or our calculations for tournaments of expected-score-maximizing agents, greater numbers of agents lead to higher winning scores, and this motivates the search for policies that have a greater chance of obtaining such scores.

8 OPTIMAL TOURNAMENT PLAY AGAINST EXPECTED-SCORE MAXIMIZING AGENTS

Finally, we turn to our central calculation, which is of policies that maximize the chance of an agent winning a tournament where all the other agents follow the strategy that maximizes expected score (MES). We calculated policies for tournaments where there are 1, 10, 100, 1000, or 10,000 other MES agents against whom one is competing. In other words, we solved the Markov decision process described in Section 3 five different times, with terminal rewards given by the five different tournament-winning probability functions shown in Figure 2B, which were derived from the distributions shown in Figure 2A. Of course, there are other tournament sizes we could have investigated, and other opponent strategies or combinations of opponent strategies we could have investigated. The single most obvious choice to us, however, was to investigate tournaments where agents are maximizing expected score.

Shortly, we will present results on how optimal tournament policies differ from the MES policy. First, however, we present some results on score distributions and chances of winning. Figure 3A shows the chance of an agent winning in tournaments where all other agents follow the MES strategy. To be clear, the agent follows a different strategy depending on the number of other MES agents in the tournament, adapting its policy to suit the tournament. For comparison, we also plot the probability of a single MES agent winning against the same numbers of other MES agents. It is clear that as tournament size increases, the chance of either a single MES agent or a tournament-adaptive agent winning decreases dramatically. Indeed, despite adapting to tournament size, the chance of winning remains at approximately one divided by the number of agents, although a growing gap can be seen between the MES and tournament-adaptive agent as tournament size increases. In Figure 3B we show the percentage increase in the chance of

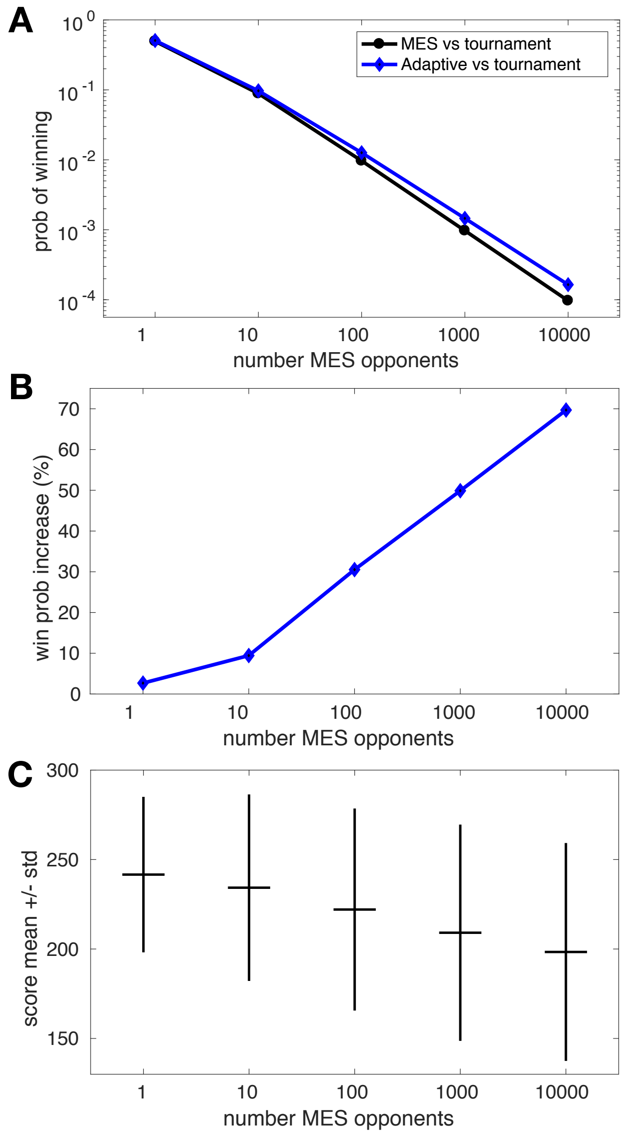

Figure 3: Statistics regarding an agent strategy that is optimal in tournaments against varying numbers of opposing agents that maximize their expected score (MES agents). (A) The chances of one MES agent or one tournament-adaptive agent winning in tournaments with different numbers of other MES agents. (B) The advantage enjoyed by a tournament-adaptive agent, in terms of percentage increase in chance of winning. (C) The mean and standard deviation of the score distributions of agents optimized for tournaments of different sizes.

winning by a tournament-adaptive agent versus the MES policy. In small tournaments, there is relatively little advantage. But in tournaments of 10,000 agents, the tournament-adaptive policy increases the chance of winning by approximately 70%—a substantial improvement. Figure 3C confirms that the tournament-adaptive agent achieves this by playing increasingly "risky" strategies as tournament size increases. The expected score of this strategy decreases with tournament size, but the variance goes up enough to increase the chance of obtaining extremely high scores, and thus winning the tournament.

Figure 4 provides a visualization of the strategy employed by a MES agent, and the optimal strategies against varying numbers of MES agents. We recognize that the strategies themselves may be of more interest to Yahtzee aficionados than to AI/agents researchers, but we feel it is valuable to demonstrate concretely how optimal strategies can change in multiagent tournaments of different sizes. In our visualization, each strategy is represented by a grid of colored rectangles, along with an additional strip just to the right of the grid. Rows of the grid correspond to different boxes in which one can score a dice roll: Ones, Twos, etc. Columns correspond to the 13 turns of the game. The color of each rectangle is determined as follows. The green intensity is proportional to the probability of scoring successfully/well in that box on that turn, whereas red intensity is proportional to the probability of cancelling/scoring poorly in the box. For the boxes Ones through Sixes, we define a successful score as one that includes three or more of the desired die. (Getting three in each box allows one to get the upper bonus of 35 points, although there are many other ways to achieve 63 points on top.) For Three-of-a-kind, Four-of-a-kind, Full-house, Small Straight, Large Straight and Yahtzee, we consider any nonzero score successful—that is, any dice roll that meets the scoring requirements of the box. Scratching the box counts as unsuccessful. For Chance, we consider a score of 18 or higher (corresponding to more than the average score of 3.5 points per die) to be successful. Thus, a red box indicates one that is likely to be scratched or scored poorly on that turn, while a green box is likely to be scored successfully on that turn. A yellow or orange box is somewhere in the middle. Within each row, we scale the intensities so that the maximum red or green intensity is 100%. Thus, different boxes within the same row can be compared directly for color or intensity, but one should not compare between rows nor between the plots for different strategies. The strip to the right of each strategy's grid displays the total success or failure probability over the course of the game.

First, we examine the MES strategy in some detail. As Figure 4 shows, that strategy is likely to succeed in the Twos through Sixes boxes, and also Three-of-a-Kind, Full House, the straights, and chance. The strategy has a respectable 33.46% change of scoring a Yahtzee at some point during the game. Perhaps most surprising is the high likelihood (89.67%) that the strategy will at some point sacrifice the Four-of-a-Kind box. The strategy is likely to score poorly in the Ones box (62.70%). A deficit in the Ones box is easily made up in other places, and this is quite consistent with how most agents play. When these plays are made are also interesting. The Yahtzee box shows the strongest trend with time. Although one cannot see because the colors are too dim, the strategy has about a 3-4% chance per turn of scoring a Yahtzee during the first five turns, and virtually no chance of scratching the Yahtzee. However, as the game approaches the final turn, the chance of scoring a Yahtzee drops to 1.05% (in part, because of the chance that the box has previously been scratched), while the chance of scratching Yahtzee on the final turn is 21.68%. Clearly, the strategy holds out hope of scoring a Yahtzee as long as it can, but will sacrifice the box if necessary—even before the final turn. This may happen, for example, if the low chance of a Yahtzee is outweighed by other important game considerations, like obtaining the top bonus. Indeed, most boxes have a rather low chance of being scored successfully if left for the last turn.

Next, we consider the optimal strategies for winning against different numbers of MES agents. We will not cover every strategy in detail, but rather focus on trends as the number of MES opponents increases. The strategy against one MES opponent appears quite similar to the MES strategy itself. Against greater numbers of MES opponents, however, the strategy is quite different. Even against just 10 other MES agents, it is very hard to win without a Yahtzee. The strategies against 10 or more MES agents focus heavily on getting one or more Yahtzees, so that the total chance in a game is over 40% (with the exact percentage varying by the number of opponents). As the number of opponents increases, the strategy is also increasingly reluctant to scratch the Yahtzee box. Indeed, against 1,000 or 10,000 opponents, the stategy will virtually only ever scratch Yahtzee on the final turn. Another major change that we see, especially against 100 or more opponents, is in how the Four-of-a-kind box is played. Although the chance of getting a Four-of-a-kind remains low, the box is not sacrificied so easily or so early. Rather, it is kept open nearly as long as possible. Perhaps one reason for this is that the strategies' desperation to get a Yahtzee makes getting a Four-of-a-kind more likely (e.g. 18.47% per game in the 10,000 opponent strategy, versus 10.33% in the MES strategy). This may also make sense because winning a game against such large numbers of opponents typically requires multiple Yahtzees in the same game. An extra Yahtzee can be scored in the Four-of-a-kind box, where it gets points both for the numbers showing on the dice and the 50 bonus points for the extra Yahtzee. Along with the increased focus on getting Yahtzees, we see an increased chance of scratching Large Straight and of scratching or scoring below average on the Ones through Sixes boxes. Indeed, in the 10,000 opponent strategy, one can almost read off a priority list of when boxes are to be sacrificed, with Large Straight often sacrificed as early as turn 9, and the Sixes and Four a Kind second only to Yahtzee in reluctance for sacrifice. In summary, we see dramatic changes in the optimal single-agent strategy depending on how many agents against which it is playing.

9 DISCUSSION

In this paper, we considered an agent playing a game in a tournament against many other agents, but without being able to observe the performance of those other agents. We proposed an optimal approach (under certain assumptions) to adapting an agent's policy for winning tournaments of different sizes, and demonstrated how radically policies could change in the domain of Yahtzee, when playing different numbers of opponents (Figures 3 and 4).

The distribution of the score-to-beat—that is, the highest score out of all opponent agents—is central to finding the best tournament policy. We proposed computing that distribution, and found that for Yahtzee, that distribution can be quite complex, with multiple modes (Figure 2). However, another possibility would be to estimate that distribution empirically (as in Figure 1), if one has access to previous

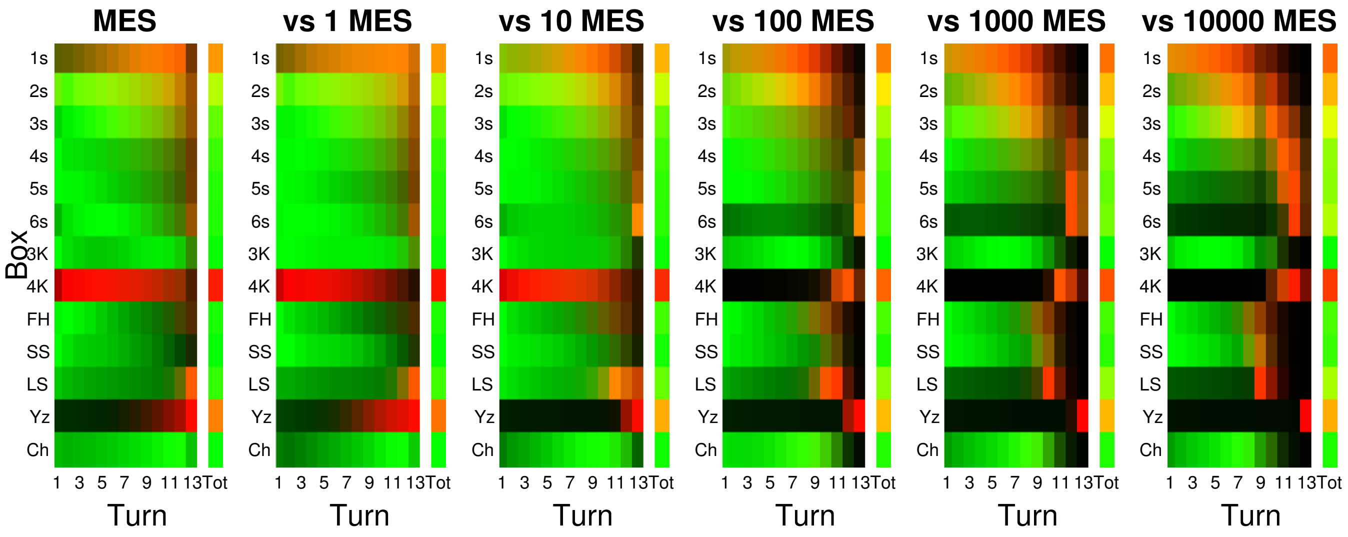

Figure 4: Comparison of optimal strategies for maximizing expected score (MES) and for maximizing the probability of winning in tournaments against different numbers of MES agents. (See text for explanation of colored plots.)

tournament results. It is not clear how well that distribution needs to be approximated, in order for the computed tournament policy to be at or nearly optimal. It may be that for many games, getting the right mean and variance is enough to generate nearly-optimal tournament play. Empirical estimation of the tournament score-to-beat distribution also avoids certain theoretical pitfalls. For example, in our analysis of Yahtzee, we assumed that opponents maximize expected score. In many tournaments, however, this is unlikely to be the case. One reason is that strict optimization of expected score is difficult. Another reason is that other opponent agents may also realize the difference between playing to maximize score and playing to win the tournament, and change their behaviour accordingly. On the other hand, explicitly reasoning about other agents' strategies, and reasoning about them reasoning about our agent's strategy, and so on, also leads to difficulties [3]. An empirical score-to-beat distribution neatly summarizes such effects without making theoretical assumptions that may be hard to justify.

Some games are too complicated to compute optimal strategies or exact score distributions. In that case, one might consider a reinforcement learning approach [18] to first compute a good single-agent policy. Then, one could use Monte Carlo sampling to estimate the distribution of winning scores for any tournament size, and finally use reinforcement learning again to compute a tournament policy. Alternatively, one might try to learn a single strategy that takes as input the game state and the tournament size, possibly represented by a deep neural network [10, 12], and train it to play tournaments of a variety of sizes. This would be very interesting because it removes the need under our scheme of recomputing a strategy for each distinct tournament size.

Another generalization of interest might be where the games the agents play are not statistically independent. For example, some human and AI-based tournaments use the same random seed for each agent/player (or the same shuffle of the deck, etc.). This reduces the influence of luck versus skill in determining the winner. In this case, convolution calculations are not a valid way of computing winning score distributions. Rather, Monte Carlo simulation of whole tournaments might be a more accurate way of estimating the score needed to win.

REFERENCES

[1] Barry C Arnold, Narayanaswamy Balakrishnan, and Haikady Navada Nagaraja. 1992. *A first course in order statistics*. Vol. 54. Siam.
[2] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. 2002. The challenge of poker. *Artificial Intelligence* 134, 1-2 (2002), 201–240.
[3] Adam Brandenburger and Eddie Dekel. 1993. Hierarchies of beliefs and common knowledge. *Journal of Economic Theory* 59, 1 (1993), 189–198.
[4] David Carmel and Shaul Markovitch. 1996. Learning models of intelligent agents. In *AAAI/IAAI, Vol. 1*. 62–67.
[5] Louis Anthony Cox, Jr. 2009. Game theory and risk analysis. *Risk Analysis: An International Journal* 29, 8 (2009), 1062–1068.
[6] CJF Cremers. 2002. How best to beat high scores in Yahtzee: A caching structure for evaluating large recurrent functions. *Master's thesis, Fac. of Math. and CS, Technische Universiteit Eindhoven, The Netherlands* (2002).
[7] Sam Ganzfried and Tuomas Sandholm. 2011. Game theory-based opponent modeling in large imperfect-information games. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 533–540.
[8] James Glenn. 2006. An optimal strategy for Yahtzee. *Loyola College in Maryland, Tech. Rep. CS-TR-0002* (2006).
[9] Christian Gollier. 2004. *The economics of risk and time*. MIT press.
[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.
[11] Vijay Krishna. 2009. *Auction theory*. Academic press.
[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.
[13] Roger B Myerson. 2013. *Game theory*. Harvard university press.
[14] Martin J Osborne and Ariel Rubinstein. 1994. *A course in game theory*. MIT press.
[15] Jakub Pawlewicz. 2010. Nearly optimal computer play in multi-player Yahtzee. In *International Conference on Computers and Games*. Springer, 250–262.
[16] Martin L Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
[17] Lloyd S Shapley. 1953. Stochastic games. *Proceedings of the national academy of sciences* 39, 10 (1953), 1095–1100.
[18] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
[19] Tom Verhoeff. 2000. Optimal solitaire Yahtzee strategies. (2000).
[20] Phil Woodward. 2003. Yahtzee®: The Solution. *Chance* 16, 1 (2003), 18–22.